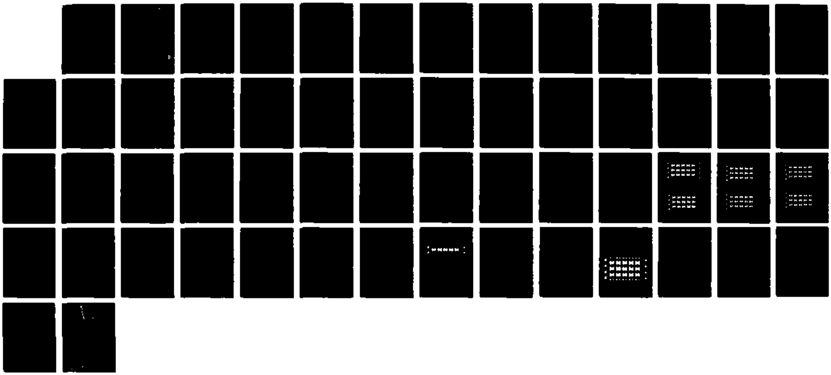
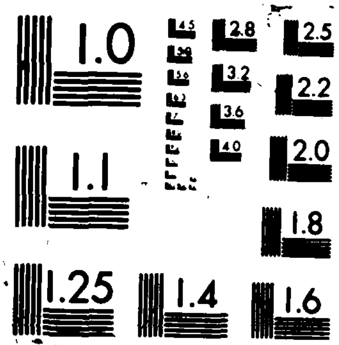
AD-A179 352   DEVELOPMENT OF A GUIDANCE SYSTEM FOR AN AUTOMATED   1/1
WAREHOUSE EQUIPPED WITH..(U) CONSTRUCTION ENGINEERING
RESEARCH LAB (ARMY) CHAMPAIGN IL   S G KAPOOR ET AL.
UNCLASSIFIED   FEB 87 CERL-TR-P-87/07   F/G 15/5   NL





MICROCOPY RESOLUTION TEST CHART

DTIC FILE COPY

USA-CERL TECHNICAL REPORT P-87/07
February 1987

2

US Army Corps
of Engineers
Construction Engineering
Research Laboratory

AD-A179 352

Development of a Guidance System for an Automated Warehouse Equipped With Mobile Robots

by
Sanjiv G. Kapoor
Robert B. Blackmon
Satheesh R. Menon
Ram Pandit
Stephen C-Y. Lu

The objective of this research was to develop a guidance system for mobile robots operating within a warehouse.

The highly structured warehouse used in model development was represented by a system of modules, alleys, and junctions which were physically identified by landmarks (e.g., patterns or bar codes mounted on the walls), and defined in the model as nodes.

Algorithms were developed for (1) path planning, (2) collision avoidance, and (3) controlling the robot's movement. Path planning occurs when a robot receives the command to deliver or retrieve an item and when it is traveling to its target module. The collision avoidance and robot movement algorithms control operations by checking the next segment of the pathway for obstacles, adjusting the robot's speed, and ensuring that the robot maintains proper orientation during movement.

The model must be extended to develop an operating system that can be easily implemented in the field.

DTIC
S ELECTE D
APR 2 7 1987
E

Approved for public release; distribution is unlimited.

87 4 24 095

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official indorsement or approval of the use of such commercial products. The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED
DO NOT RETURN IT TO THE ORIGINATOR

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704 0188
Exp Date Jun 30 1986

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS
A179352

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
USA-CERL TR P-87/07

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
Department of Mechanical and Industrial Engineering

6b OFFICE SYMBOL *(If applicable)*

7a NAME OF MONITORING ORGANIZATION
U.S. Army Construction Engr. Research Lab.

6c ADDRESS *(City, State, and ZIP Code)*
University of Illinois
Champaign, IL 61820

7b ADDRESS *(City, State, and ZIP Code)*
P.O. Box 4005
Champaign, IL 61820

8a NAME OF FUNDING / SPONSORING ORGANIZATION
U.S. Army Construction Engr. Research Lab.

8b OFFICE SYMBOL *(If applicable)*
USA-CERL

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS *(City, State, and ZIP Code)*
P.O. Box 4005
Champaign, IL 61820

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE *(Include Security Classification)*
Development of a Guidance System for an Automated Warehouse Equipped with Mobile Robots

12 PERSONAL AUTHOR(S)
KAPOOR, Sanjiv G; BLACKMON, Robert B; MENON, Satheesh R; PANDIT, Ram; LU, Stephen C-Y

13a TYPE OF REPORT
Final

13b TIME COVERED
FROM ________ TO ________

14 DATE OF REPORT *(Year, Month, Day)*
87-02

15 PAGE COUNT
55

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB GROUP
12	09	

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Warehouse
Guidance
Robot

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

>The objective of this research was to develop a guidance system for mobile robots operating within a warehouse.

The highly structured warehouse used in model development was represented by a system of modules, alleys, and junctions which were physically identified by landmarks (e.g., patterns or bar codes mounted on the walls), and defined in the model as nodes.

Algorithms were developed for path planning, collision avoidance, and controlling the robot's movement. Path planning occurs when a robot receives the command to deliver or retrieve an item and when the robot is traveling to its target module. The collision avoidance and robot movement algorithms control operations by checking the next segment of the pathway for obstacles, adjusting the robot's speed, and ensuring that the robot maintains proper orientation during movement.

(Cont'd on back)

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
D.P. Mann

22b TELEPHONE *(Include Area Code)*
(217) 373-7223

22c OFFICE SYMBOL
Chief-IMT

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

BLOCK 19 (Cont'd)

The model must be extended to develop an operating system that can be easily implemented in the field.

FOREWORD

This project was funded by the Facility Systems (FS) Division of the U.S. Army Corps of Engineers Research Laboratory (USA-CERL) under the Technical Director's initiative. The work was accomplished under contract by the Department of Mechanical and Industrial Engineering at the University of Illinois at Urbana/Champaign. The researchers were Sanjiv G. Kapoor, Satheesh R. Menon, Ram Pandit, and Stephen C-Y. Lu. The Principal Investigator and contract monitor was Robert B. Blackmon. Mr. Ed Lotz is the Chief of FS.

COL Norman C. Hintz is Commander and Director of USA-CERL and Dr. L.R. Shaffer is Technical Director.

Accession For
NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes

Dist	Avail and/or Special
A-1	

QUALITY INSPECTED 1

CONTENTS

TABLES

FIGURES

DEVELOPMENT OF A GUIDANCE SYSTEM FOR AN AUTOMATED WAREHOUSE EQUIPPED WITH MOBILE ROBOTS

1 INTRODUCTION

Background

In case of national mobilization, the U.S. Army must be able to construct large training camps within a very short time with readily available materials, and with a severe shortage of skilled labor. A major investment has been made in examining the facility delivery system, evaluating alternative types of facilities, developing planning tools, and investigating the capability of private industry to help meet this requirement. A standard set of building designs has been developed to eliminate much of the normal effort required to prepare construction contracting documents. However, this effort has not resulted in decreased construction time nor reduced the need for skilled labor.

The Army faces a similar problem when constructing facilities in the theater of operations. Troop labor is used to construct semipermanent facilities to replace tents and other temporary shelters.

Incorporating robots into the construction process appears to be the optimal solution. Intelligent equipment would lessen the need for large skilled and semiskilled labor pools, eliminate many of the repetitive and boring tasks which often lead to accidents, and simplify many construction activities.

This study addresses the possibility of using robots to operate a warehouse at a construction site. A fully automated warehouse could easily be operated by a small crew of workers to supervise operations and maintain the equipment. In addition to reducing manpower requirements, this system of robots and computers could simplify inventory control, shorten delivery times, and decrease worker exposure to accidents.

Palletized materials and equipment would be picked up by a forklift upon arrival on the receiving platform. As the vehicle passes into the warehouse, sensors on the door frames could read the bar codes on the pallet and increase the inventory. As the pallet is eventually moved to the loading dock, the inventory could be automatically reduced by the same method. Each storage area would be controlled by a central computer to assist in the initial path planning and monitor movements of robots in the area. The same computer could keep track of where all the supplies are stored. The use of a central computer to locate supplies for automated retrieval eliminates the need for a highly structured warehouse layout and segregation of supplies by part numbers. Supplies could be stored on a space-available basis with little formal structure.

This study addresses the problems of operating mobile robots within a defined facility using master floor plans stored in the robot's on-board computer and landmarks for guidance. This technology will then serve as the basis for addressing the problem of managing self-navigating robots traveling between the warehouses and erection sites.

Mobile systems have been built for navigation in a well-defined domain such as a house, office, or factory floor.[1] These systems employ stereoscopic visual sensors, ultrasonic range finders, and laser range finders for guidance. They continually create local maps and update their position on the global map. Path tracking is accomplished by receiving signals through embedded wire or by following tapes laid on the floor. At present, the mobile systems perform simple and limited tasks. The systems tend to break down as the tasks become complicated, the pathways become more cumbersome, and the number of robots working in the same space increases.

Purpose

The purpose of this research was to develop a guidance system for mobile robots operating within a warehouse.

Approach

Research was conducted in three stages. Stage I, covered in Chapter 2, dealt primarily with the planning and design of the warehouse and its components, and the development of a landmark system for robot guidance.

Path planning algorithms were developed in Stage II, as explained in detail in Chapter 3. Two algorithms were formulated for selecting the optimum path for a single robot and later generalized to manage multiple robots. The algorithms for controlling robot movement incorporate collision avoidance strategies.

To demonstrate the effectiveness of the algorithms, an IBM PC-based animated color graphics simulation package was created in Stage III. The package is user friendly and allows both static and dynamic simulation as would be expected in an actual working environment. Chapter 4 describes development of the simulation model.

Chapters 5 and 6, respectively, contain the summary of the research effort at the end of the first year, and recommendations for future projects.

Assumptions

The field warehouse will have a concrete floor but it will not have the smoothness of factory floors finished for robotic traffic.

The warehouse and storage yard will be sized for three complete facilities. Supplies and materials will be arriving on a phased basis approximating the need for supplies at the erection site. The number of pallets to be handled and stored was based on an analysis of one facility as described in Chapter 2.

[1] J. L. Crowley, "Navigation for an Intelligent Robot," *IEEE Journal of Robotics and Automation*, March 1985, pp 31-41; S. C-Y. Lu, T. D. Brand, and S. G. Kapoor, *Sensor and Guidance Technology Related to Mobile Robot Devices*, Technical Report P-87/02 (U. S. Army Construction Engineering Research Laboratory [USA-CERL]).

The robot vehicle is assumed to resemble a cylinder with standard forks and lift mechanism. More details are given in Appendix A.

Initial path planning will be accomplished by the central computer. The central computer will maintain the inventory, relate stock items to locations, and record the locations of known obstacles. As the robot starts on its path, an on-board computer assumes the path planning and guidance responsibilities using landmarks.

An operational concept has been assumed in this study as the basis for developing the control algorithms; however it can be modified to comply with local requirements and facility layouts.

Mode of Technology Transfer

Upon completion of the final algorithms and definition of the optimal landmarks, the control system will be tested and demonstrated at an Army warehouse using commercially available mobile robots modified to accept the sensors and more complex programming developed in this project. Decisions on field implementation and industry involvement in producing the needed mobile robots will be based on the outcome of the field test.

2 WAREHOUSE PLANNING

In developing a guidance system for an automated warehouse, the first step was to design the layout and the landmark system. Landmarks are special signs or patterns used for identification. These could be bar codes or objects of different shapes located near the entrances of modules, alleys, or junctions. The information contained in the landmarks includes module number, junction number, and direction of movement. The positioning of landmarks should provide maximum information using a minimum number of landmarks. The layout was based on modularity, flexibility, and quick installation.

Description of the Warehouse

The warehouse design was based on the materials data for building construction supplied by the U.S. Army Construction Engineering Research Laboratory (USA-CERL). The warehouse contains sufficient quantities of each item to serve three to four facilities at a time. Table 1* lists the items and number of pallets required for one facility.

The warehouse design was assumed to be composed of several modules. The main storage area has an overall dimension of 480 ft by 300 ft.** There are five modules in a row and eight in a column. All traffic lanes are 5 ft wide. The other main components that make up the warehouse include blocks, junctions, mainway, highways, freeways, alleys, and bypasses which will be described in detail in the following paragraphs.

Module

A standard module (60 ft by 60 ft) is depicted in Figure 1. Each module consists of four aisleways. Articles can be stored on either side of the aisleways. The module has an entrance and an exit (identified as IN and OUT, respectively). Inside each module, the traffic moves in a counterclockwise direction as represented by arrows.

The four different pathways in a module are:

1. Aisleways: The aisleways have a "working area," 20 ft long in the middle and waiting "sheds" 5 ft long on both ends. The pick and place operation is performed in the working area. Since only one robot is allowed into the working area at any time, other robots must wait in the shed.

2. Freeway: The freeway goes all around the aisles, allowing traffic to enter and depart the aisleways. Also, since it forms a complete loop around the aisles, any robots that are denied the chance to enter any one of the aisleways can go around and try again.

3. Bypass: The bypass provides an alternate route for a robot which does not have to visit that particular module. The direction of motion is shown by arrows (from IN towards OUT).

4. Alleys: There are two alleys for each module: one near the entrance and one near the exit. Traffic moves toward the exit OUT in alley AL2 and away from the entrance IN in alley AL1.

*Tables and Figures are located at the end of each chapter.

**1 ft = 0.305 m.

Note that the hatched blocks near the alleys are only separators but can be used to provide landmarks for module information. The landmarks are symbolically represented by a string of X's.

For easy storage, the items are palletized. Bar codes may be used to identify pallets. The standard pallet is 4 ft long, 3 ft deep, and 2 ft high. However, to accommodate items of varying heights, the pallet heights can vary. The pallets are arranged lengthwise (4 ft side) on either side of the working area of each aisleway so a maximum of five pallets will fit on one side of the aisleway (5 pallets, 4 ft each). The levels of storage depend on the pallet heights. Based on the maximum lift capability in the robots, three levels of standard-height pallets can be stored.

Assigning building supplies to the modules is done arbitrarily (Table 2). The items assigned to a module, whether they be only of one kind or different kinds, are equally distributed among the four aisleways within each module. The robots entering the module will then have the choice to visit any one of the four aisleways instead of circling on the freeway or waiting in the shed until the working area is clear. This provides flexibility within the module.

The robot traveling along the bypass senses the module information near the entrance. If the robot has to visit the module, it enters the freeway. Otherwise it proceeds along the bypass to the next module. Once in the freeway, the robot moves to the first aisle it encounters. If there is already a robot waiting in the shed, it moves to the next aisle entrance. Note that a module can take care of eight robots at any time: four actually working in the aisles and four waiting in the sheds near the aisle entrances. Since the freeway forms a loop around the aisles, additional robots can travel along the freeway loop and search continuously for an empty aisle.

Blocks

The pick and place operation can be made much faster if alternate ways are found for the robot to reach its desired module and get out of the storage area. By classifying a group of modules into blocks, the robot can search for its target module without searching the entire storage area. As mentioned before, modules are arranged in rows of five. A group of two rows of modules which face each other (Figure 2) comprises a block. The blocks are identified as A-AA, B-BB, C-CC, and D-DD (Figure 3).

Mainways

When the modules are packed side by side to form a row, their bypasses are joined, end to end, to form one of the lanes of the mainway. Thus, in a block, the mainway is the two-lane pathway formed by the adjacent bypasses. Traffic moves in opposite directions in the lanes as indicated by arrows in Figure 2.

When two blocks intersect as shown in Figure 4, the backs of the modules are attached. The alleys of the intersecting modules are joined to provide an alternate path for the robot to go from one block to another.

Main Exits and Inlets

Figure 3 shows the layout of the overall warehouse. Various exits are provided on either side of the storage area. The receiving and shipping dock locations are flexible.

Highways and Junctions

Highways are the high-speed lanes for the robot to reach the desired blocks. They have two lanes with traffic flowing in opposite directions.

The mainways leading from the blocks intersect with the highways at junctions. The traffic control at the junctions is presented in detail later in this report.

Labeling System

The junctions are labeled A, B, C, D, AA, BB, CC, and DD. Modules that could be accessed through junctions AA, BB, CC, and DD are even-numbered from right to left and those that could be accessed through A, B, C, and D are odd-numbered from left to right. For example, as shown in Figure 2, the upper modules are numbered 2, 4, 6, 8, and 10 from right to left and the lower modules that could be assessed through entrance A are numbered 1, 3, 5, 7, and 9 from left to right. Note that this numbering system was adopted only for easy module identification and could be replaced with any other numbering system.

Sensor Requirements

Sensors are required for the robot to grasp the information provided by the landmarks and plan its course of action. A survey on sensors was carried out in previous work.[2] The following section deals with the specific requirements that the sensor must satisfy for module identification, block identification and traffic control.

Module Identification

Information identifying the module should be provided near the entrance of each module on the mainway (bypass) and on the alleys near the entrance (Figure 1). Sensors provided near the exit of each module on the mainway (bypass) give information about the module in the same column that could be visited by moving through the alley (Figure 1).

Block Identification

Blocks are identified as the robot approaches the junctions near the block entrance on the highways. Block information may be provided using patterns or bar codes. Blocks are made up of two rows of modules.

Traffic Control at the Junctions

A typical junction on a highway is shown in Figure 5. To explain the working, the right lanes of the four crossroads that form the junction are numbered, 1, 2, 3, and 4. The left lanes, right turns, straight move, and halt/vacant mode are represented as L, R, S, and H, respectively.

Table 3 summarizes a list of motion rules that are permitted to happen simultaneously. For instance, rule 4 allows 1S, 2H, 3S, and 4R to occur at the same time. In other words, if the robot in lane 1 wants to go forward, then the robot in lane 2

[2] S. C-Y. Lu, T. D. Brand, and S. G. Kapoor.

will have to halt while lane 3 robot is allowed to go straight only and lane 4 robot to make a right turn. When a rule is executed and the robot in any lane has a different motive than the allowed motion, it merely halts.

The sensor provided at the junction checks the status of lanes 1, 2, 3, and 4. In other words, the robots in lanes 1, 2, 3, and 4 communicate with the junction sensor about their motives. The sensor then takes the status of one of the lanes and compares it with the list of rules. If there is more than one allowed rule for a particular mode in a particular lane (e.g., there are two allowed rules for a robot in lane 1 with a status mode S), one rule is arbitrarily selected. The sensor waits for a definite interval of time for the rule sequence to be executed. The time lag is set depending on the maximum time to execute a motion.

The sensor then checks the status of the next lane and compares it with the set of rules. If there is no robot waiting at a lane, the sensor merely assigns the lane status as halt mode H and checks for the set of rules allowed with H modes in that lane. Rules which are not chosen earlier will be preferred over the selected ones when there is more than one set of allowed rules.

Traffic inside the storage area is controlled mainly by avoiding collisions and alternate path planning. Once inside the module, the robot has the option of entering any of the four aisleways, provided there is no robot waiting in the shed. If all the aisles are full, it can travel around on the freeway until an aisle is clear. Beacon sensors or similar type sensors are required at the exits of the module to allow right turns to join the traffic in the mainway. The main characteristics and the desired functions of the sensors are summarized in Table 4.

Table 1

Item Description and Number of Pallets Required

No.	Description	Units	Dimensions, in.	Items per Pallet*	No. of Pallets
1.	Asphalt felt	63	8 x 8 x 36	16	4
2.	Strip shingles	25	48 x 48 x 43	1	25
3.	Painting	22	20 x 20 x 10	6	4
4.	Strap ties to roof	5	12 x 24 x 8	5	1
5.	1/2 x 4 lag boxes	9	4 x 4 x 5	9	1
6.	Paint dry wall	96	20 x 20 x 10	8	12
7.	Vat floor	534	12 x 12 x 6	45	12
8.	Vapor burner	20	6 x 6 x 48	20	1
9.	Fire extinguishers	19	8 x 8 x 20	19	1
10.	Soap dishes	1	24 x 24 x 24	1	1
11.	Towel pegs	1	24 x 24 x 24	1	1
12.	Mirrors	48	24 x 30 x 2	24	2
13.	Mirrors, 24" convex	2	24 x 24 x 2	2	1
14.	Toilet paper holder	32	6 x 6 x 6	32	1
15.	Wood shelves	1	10 x 32 x 48	1	1
16.	Door safety glass	1	30 x 36 x 8	1	1
17.	Door hardware bolt and hinges	3	12 x 15 x 12	3	1
18.	Door closers	6	12 x 15 x 12	6	1
19.	Door sets	2	12 x 15 x 12	2	1
20.	Door latch sets	30	12 x 15 x 12	30	1
21.	Thresholds	1	24 x 24 x 38	1	1
22.	Ells	18	12 x 12 x 15	9	2
23.	Couplings	9	12 x 12 x 15	9	1
24.	Gate valves	5	12 x 12 x 15	5	1
25.	HW Circulation Pump 40 gpm	1	8 x 8 x 16	1	1
26.	HW Circulation Pump 10 gpm	1	8 x 8 x 16	1	1
27.	Exhaust fan	20	14 x 14 x 8	10	2
28.	Thermostat wire	11	12 x 12 x 15	11	1
29.	3-2 service entrance	1	30 x 30 x 29	1	1
30.	2-12 w/Romex	77	16 x 16 x 4	26	3
31.	3-12 rolls	16	24 x 24 x 4	8	2
32.	Wo12 THE	1	12 x 12 x 4	1	1
33.	1/10 THE	1	24 x 24 x 4	1	1
34.	256 MCM THE	2	30 x 30 x 4	2	1
35.	Duplex outlets	20	48 x 5 x 6	20	1
36.	Switches	11	48 x 5 x 6	11	1
37.	Insulation pipe, 4"	5	12 x 12 x 48	5	1
38.	Insulation pipe, 3"	12	12 x 12 x 48	6	2
39.	Insulation pipe, 2.5"	2	12 x 12 x 48	2	1
40.	Insulation pipe, 2"	10	12 x 12 x 48	5	2
41.	Insulation pipe, 1.5"	7	12 x 12 x 48	4	2

*Standard pallet size: 4 x 3 x 2 ft.

Table 1 (Cont'd)

No.	Description	Units	Dimensions, in.	Items per Pallet	No. of Pallets
42.	Insulation pipe, 1.25"	3	12 x 12 x 48	3	1
43.	Insulation pipe, 1"	2	12 x 12 x 48	2	1
44.	Insulation pipe, 3/4"	5	12 x 12 x 48	5	1
45.	Insulation pipe, 1/2"	1	12 x 12 x 48	1	1
46.	Drinking fountain	4	20 x 20 x 30	2	2
47.	Tees	15	12 x 12 x 15	15	1
48.	GFI	1	12 x 12 x 15	1	1
49.	Junction bars	9	16 x 24 x 10	5	2
50.	Light fixtures	298	16 x 16 x 10	8	36
51.	Engineering lights	28	12 x 12 x 8	28	1
52.	Thermostats	11	12 x 15 x 12	11	1
53.	Ceiling Detectors	1	24 x 24 x 12	1	1
54.	Manual stations	1	12 x 15 x 12	1	1
55.	Arc lamps	1	12 x 12 x 8	1	1
56.	Electric panels	6	16 x 30 x 6	1	1
57.	Window frames	370	36 x 42 x 6	4	93
58.	Glazing	4	30 x 24 x 12	4	1
59.	Grills	20	16 x 16 x 20	4	5
60.	Ells	1	12 x 15 x 12	1	1
61.	Tees	2	12 x 15 x 12	2	1
62.	Valves	2	12 x 15 x 12	2	1
63.	Flue pipe, 14	7	16 x 16 x 48	2	4
64.	Flue pipe, 12	77	14 x 14 x 48	2	39
65.	Flue pipe, 10	34	12 x 12 x 48	6	6
66.	Flue pipe, 8	42	10 x 10 x 48	6	7
67.	Flue pipe, 7	20	8 x 8 x 48	10	2
68.	Ridge vents	20	8 x 8 x 10'	20	1
69.	Panel siding	8	48 x 35 x 8'	1	8
70.	Caulking	15	8 x 10 x 14'	15	1
71.	Metal sheets	1	48 x 24 x 8'	1	1
72.	J. molding	4	4 x 4 x 10'	1	4
73.	Plastic wall	5	48 x 35 x 8'	1	5
74.	Sill window	1	6 x 6 x 16'	1	1
75.	Bench	1	16 x 12 x 8'	1	1
76.	1021 HWT	1	48 x 48 x 11'	1	1**
77.	Conduit (½" and 2½")	1	4 x 4 x 10'	1	1
			6 x 6 x 10'	1	1
78.	Door frames, 3'	52	40 x 5 x 7'	6	9
79.	Door frames, 6'	1	76 x 5 x 7'	1	1**
80.	Doors, 3'	59	36 x 1.5 x 80	30	2
81.	Doors, 6'	2	36 x 1.5 x 80	30	1
82.	Wall, 5/8"	53	48 x 35 x 8'	1	53
83.	Corner bead	1	6 x 6 x 8'	1	1
84.	Hot water tank, 420 BTU	2	30 x 30 x 7'	2	2
85.	Gas furnace	10	30 x 30 x 6'	2	5
86.	Plywood, 1/2"	16	48 x 35 x 8'	1	16

**Indicates pallets of odd dimensions.

Table 1 (Cont'd)

No.	Description	Units	Dimensions, in.	Items per Pallet	No. of Pallets
87.	Plywood, 3/4"	16	48 x 33 x 8'	1	16
88.	Lumber, 2 x 4 studs	17	45 x 33 x 8'	1	17
89.	Lumber, 2 x 4s	40	42 x 24 x 16'	1	40
90.	Lumber, 2 x 8s	1	11 x 24 x 16'	1	1
91.	Lumber, 2 x 6s	1	45 x 27 x 16'	1	1
92.	Lumber, 1 x 4s	1	3 x 15 x 16'	1	1
93.	Trim, 1 x 6s	1	24 x 27 x 16'	1	1
94.	Wood base	19	5 x 7 x 16'	19	1
95.	Stair strips	1	48 x 24 x 16'	1	1
96.	Lintels	1	42 x 6 x 16'	1	1
97.	Blocking	1	42 x 12 x 16'	1	1
98.	Hot water heater storage tank	1	48 x 48 x 11'	1	1**
99.	Pipes, cu				
	(a) 4", 3", 2.5"	1	36 x 36 x 20'	1	1
	(b) 2", 1.5", 1.25"		30 x 30 x 20'	1	1
	(c) 1", 4.76", 0.5"		36 x 36 x 20'	1	1
100.	Gas Pipes				
	(a) 4", 3", 2.5" 1.25", 1"	1	38 x 21 x 20'	1	1
	(b) 2", 3"	2	30 x 30 x 20'	1	2

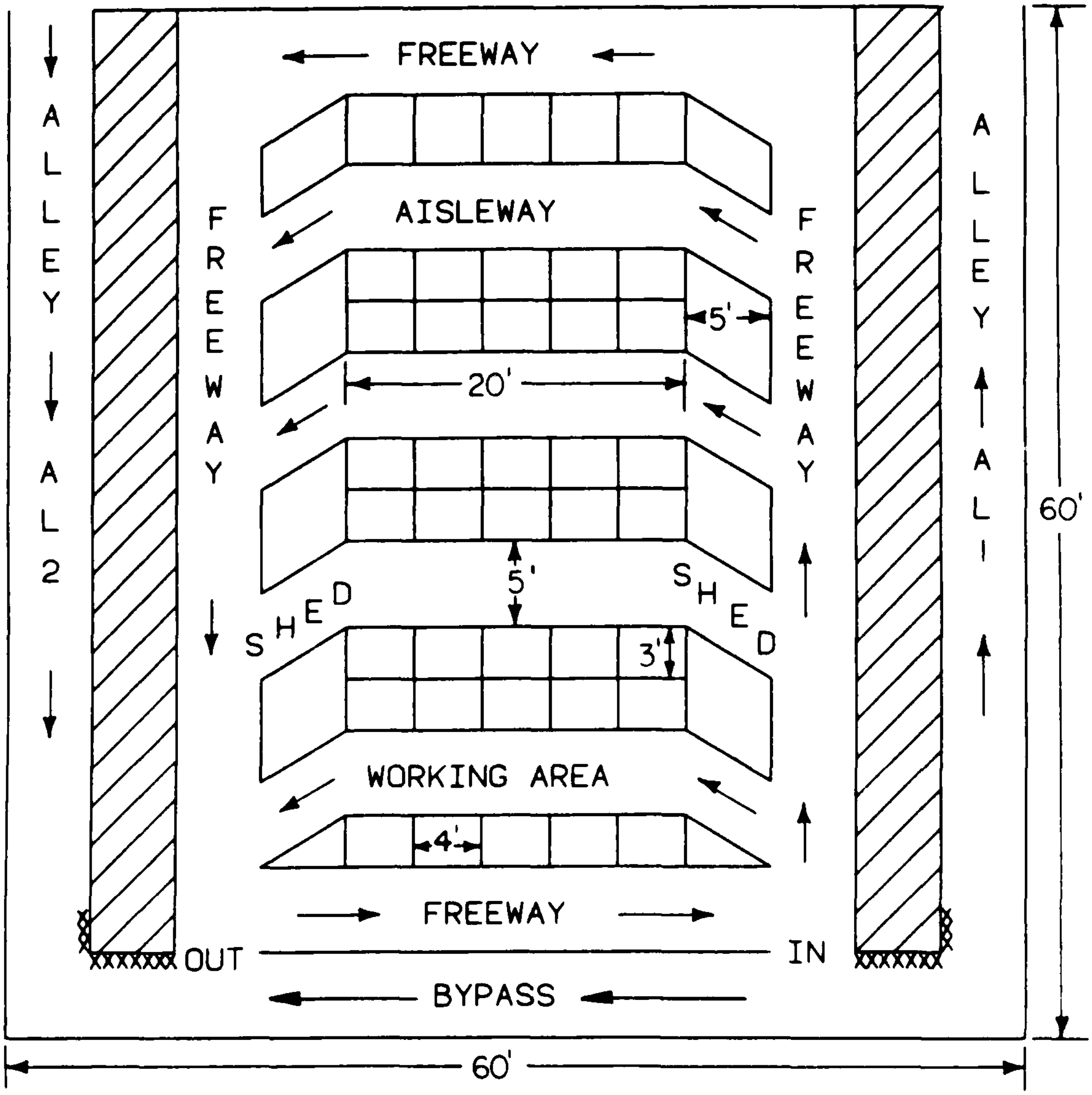


Figure 1. Module.

Table 2

Assignment of Items to Modules

Module No.	Item No.*	Items in One Aisle	Levels on One Side	Length Occupied on One Side (ft)
A1	1	6	3	4
	3	6	3	4
	6	12	3	8
	63	6	3	4
A2	7	12	3	8
	12	6	3	4
	22	6	3	4
	27	6	3	4
A3	50	30	3	20
A4	64	30	3	20
A5	65	6	3	4
	66	6	3	4
	59	6	3	4
	30	6	3	4
	21	6	3	4
A6	39	4	2	4
	31	4	2	4
	40	4	2	4
	41	4	2	4
	46	4	2	4
A7	57	30	3	20

*See Table 1 for item description.

Table 2 (Cont'd)

Module No.	Item No.	Items in One Aisle	Levels on One Side	Length Occupied on One Side (ft)
A8	57	30	3	20
A9	57	30	3	20
A10	57	30	3	20
B1	4	2	1	4
	5	2	1	4
	8	2	1	4
	9	2	1	4
	10	2	1	4
B2	17	2	1	4
	18	2	1	4
	19	2	1	4
	20	2	1	4
	23	2	1	4
B3	24	2	1	4
	25	2	1	4
	26	2	1	4
	28	2	1	4
	29	2	1	4
B4	32	2	1	4
	33	2	1	4
	34	2	1	4
	35	2	1	4
	36	2	1	4
B5	37	2	1	4

Table 2 (Cont'd)

Module No.	Item No.	Items in One Aisle	Levels on One Side	Length Occupied on One Side (ft)
B5	39	2	1	4
(Cont,d)	42	2	1	4
	43	2	1	4
	44	2	1	4
B6	45	2	1	4
	47	2	1	4
	48	2	1	4
	49	2	1	4
	50	2	1	4
B7	51	2	1	4
	52	2	1	4
	53	2	1	4
	54	2	1	4
	55	2	1	4
B8	56	2	1	4
	58	2	1	4
	60	2	1	4
	61	2	1	4
	62	2	1	4
B9	73	4	1	8
	75	2	1	4
	83	2	1	4
	68	2	1	4
B10	86	10	1	20
C1	86	6	1	12

Table 2 (Cont'd)

Module No.	Item No.	Items in One Aisle	Levels on One Side	Length Occupied on One Side (ft)
C1	85	2	1	4
(Cont'd)	84	2	1	4
C2	87	10	1	20
C3	87	6	1	12
	80	2	1	4
	81	2	1	4
C4	88	10	1	20
C5	88	8	1	16
	77	2	1	4
C6	89	10	1	20
C7	89	10	1	20
C8	89	10	1	20
C9	89	10	1	20
C10	70	2	1	4
	72	2	1	4
	74	2	1	4
	90	2	1	4
	91	2	1	4
D1	78	8	1	16
	92	2	1	4

Table 2 (Cont'd)

Module No.	Item No.	Items in One Aisle	Levels on One Side	Length Occupied on One Side (ft)
D2	93	2	1	4
(Cont'd)	94	2	1	4
	95	2	1	4
	96	2	1	4
	97	2	1	4
D3	99a	2	1	4
	99b	2	1	4
	99c	2	1	4
	100a	2	1	4
	100b	2	1	4
D4	82	10	1	20
D5	82	10	1	20
D6	82	10	1	20
D7	82	10	1	20
D8	76	2	1	
	79	2	1	(odd aisles)
		2	2	1
	98	2	1	
D9	69	8	1	16
	71	2	1	4
D10	11	2	1	4
	13	2	1	4
	14	2	1	4

Table 2 (Cont'd)

Module No.	Item No.	Items in One Aisle	Levels on One Side	Length Occupied on One Side (ft)
D10	15	2	1	4
(Cont'd)	16	2	1	4

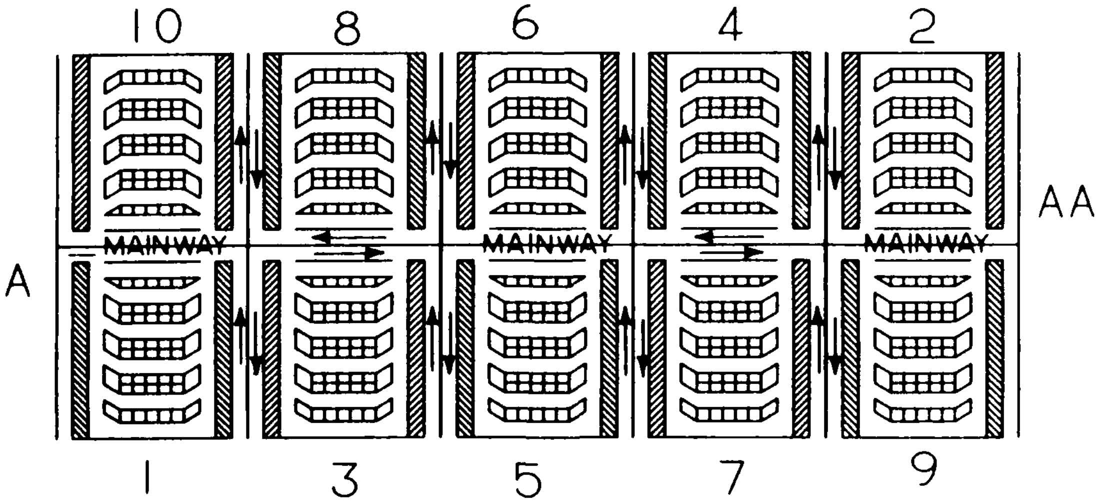


Figure 2. Block and mainway.

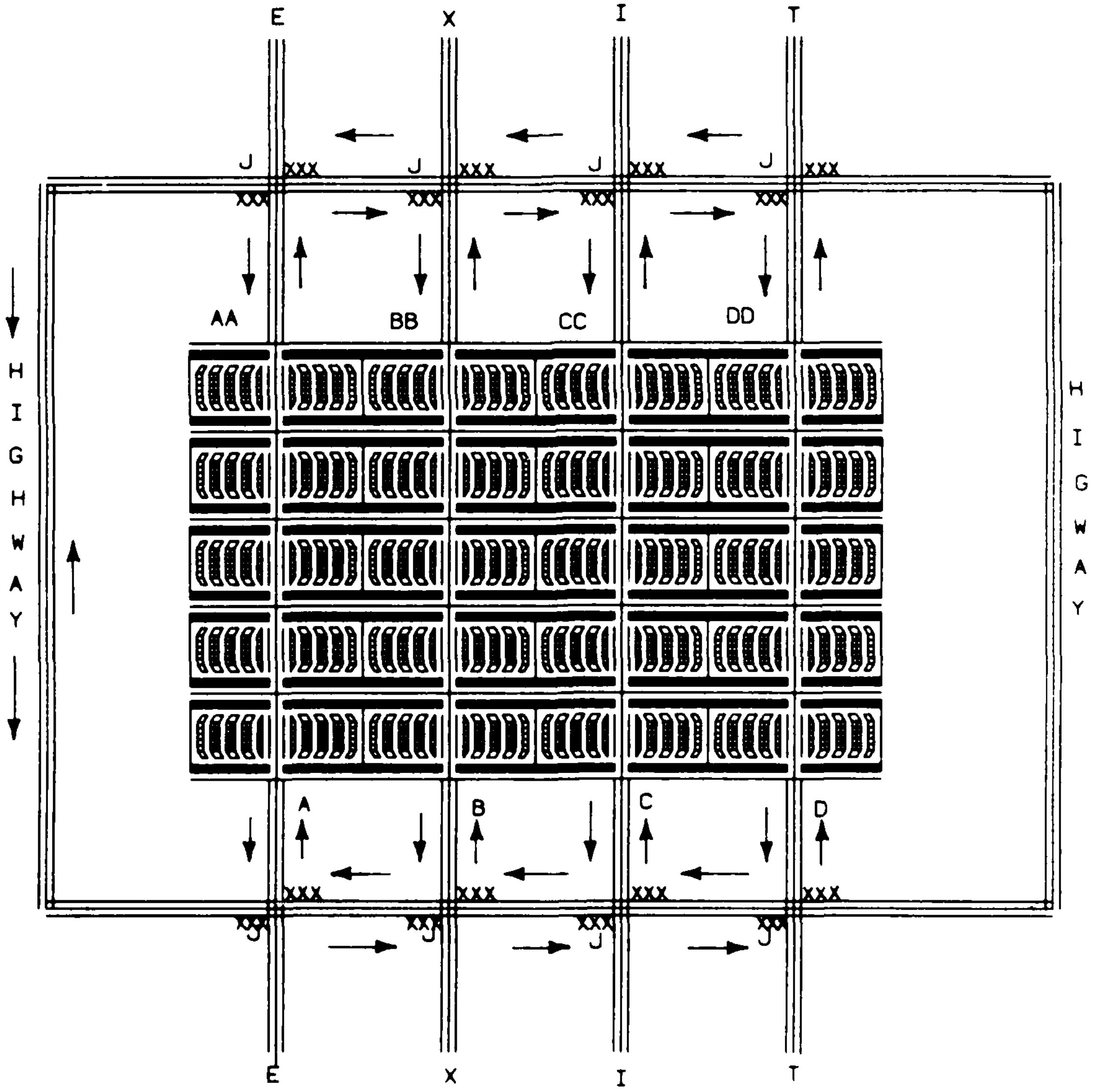


Figure 3. Warehouse layout.

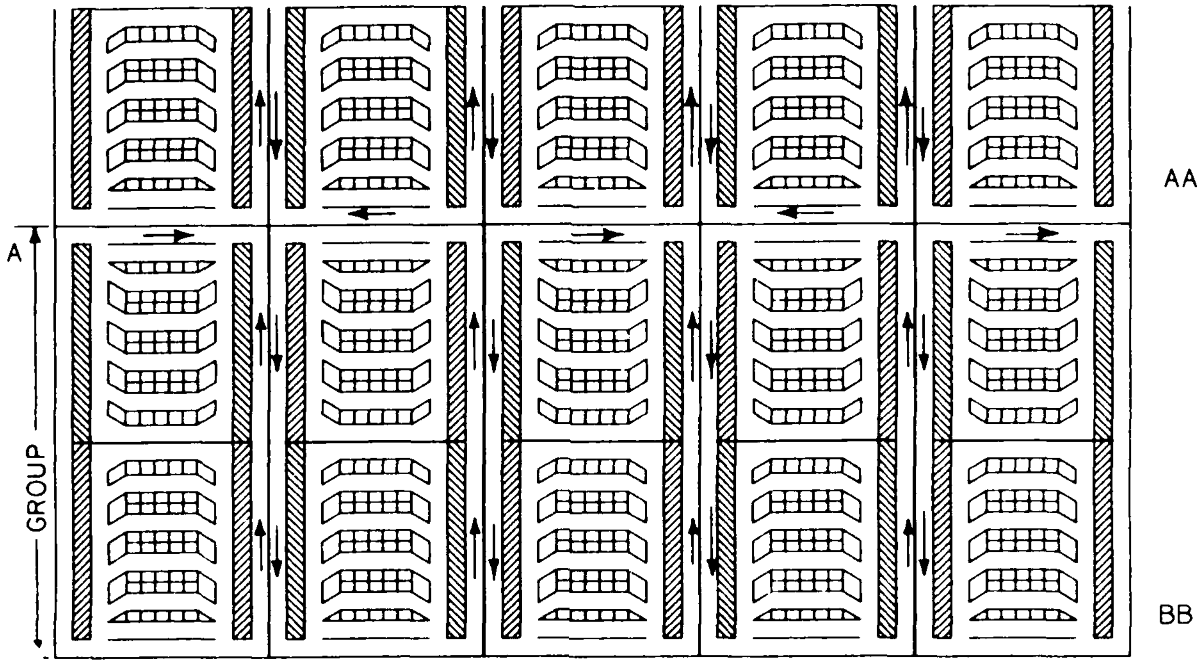


Figure 4. Movement along alleys.

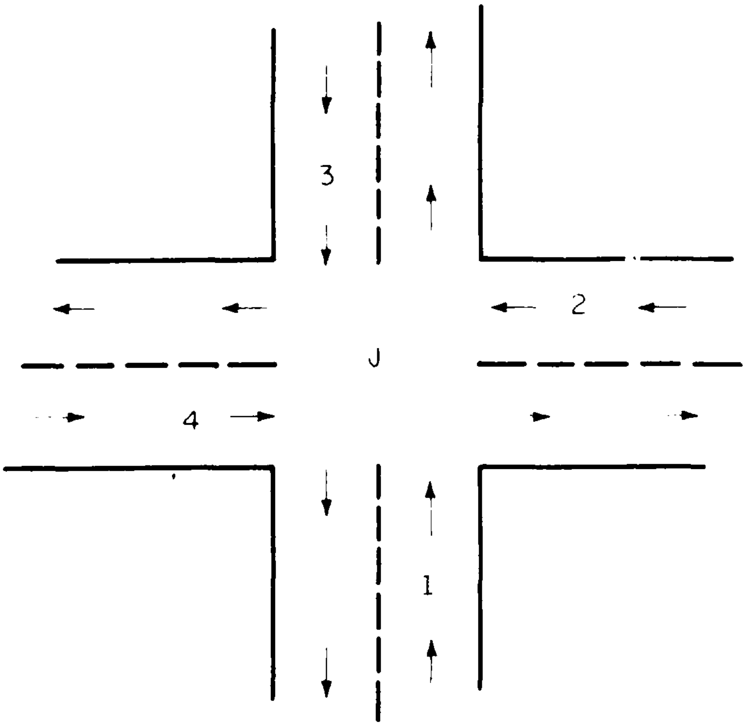


Figure 5. Traffic control at a junction.

Table 3

Allowed Motion Sequence at Junction

Rule No.	Allowed Motion Sequences			
1	1H	2S	3H	4S
2	1H	2H	3R	4L
3	1H	2R	3L	4H
4	1S	2H	3S	4H
5	1S	2H	3R	4R
6	1R	2R	3S	4H
7	1R	2R	3R	4R
8	1R	2S	3H	4R
9	1R	2L	3H	4R
10	1L	2H	3H	4R

Table 4

Characteristics and Functions of the Sensors

Modules

Decode module information from landmarks near the module entrances
Identify IN and OUT
Read codes on pallets
Maintain distance between robots and between robots and obstacles
Allow merging from freeway into aisleways and from freeway into mainway (bypass)

Blocks

Decode block information from landmarks near the junction on highways
Allow merging from mainway into freeway

Junction

Check the status of the right lanes of the four-cross-roads at definite time intervals
Communicate with the robots in the four lanes

Pathways

Maintain position of robots along the central region of the lanes, thereby providing linear motion all the time (a simple compass type sensor may be used here)

3 PATH PLANNING

Most of the path planning algorithms developed for mobile robots[3] are meant for situations where the robot is required to navigate in an unexplored environment. To understand its environment, a mobile robot should be able to consistently model the environment and to locate itself correctly.[4] Sensors help the robot define reference landmarks while exploring the environment. The world model created by the robot is called the local map. The original world model of the entire environment is called the global map. Position referencing is done by comparing the local map with the global map. In this way, as the robot cruises the unexplored environment, it creates its own map. The next step is to plan its path to the target position. One of the commonly employed path planning routines is the path relaxation method.[5] In this method, the global model is covered with a grid of connected points. The robot chooses a rough path along the nodes of the grid based on the "cost" of each node. The node cost is composed of the cost for distance, near objects, and for being within or near an unmapped region. The optimal path is found using an A* search. The chosen path is then "relaxed" (the nodes are moved on the path) to minimize the total cost. This model does not assume a completely known world with planar-faced objects.

The design of the warehouse layout and the positioning of the landmarks are such that the robot is able to understand the environment without creating a local map. The number of different paths available for the robot to move towards the target module is limited. Note that this limitation is due to the imposed direction of movement along the mainways and highways (Figure 3). The objectives used to optimize the selected path include minimum time path and shortest path. The idling robot is required to select the quickest time path while a loaded robot should choose the shortest path available.

Primary Considerations

1. The mean velocity of the robots on the highway is much higher than on the mainways. Hence, it is desirable for the robots to prefer movement along the highway to reach the target module.

[3] R. A. Brooks, "Solving the Find Path Problem by Good Representation of Free Space," *International Journal of Robotics and Automation, IEEE* (1983), pp 190-197; J. L. Crowley, pp 31-41; L. Gouzenes, "Strategies for Solving Collosion-free Trajectories Problems for Mobile and Manipulation Robots," *The International Journal of Robotics Research*, Vol 3, No. 4 (1984), pp 51-64; O. Katib, "Real-Time Obstacle Avoidance for Manipulator and Mobile Robots," *IEEE International Conference on Robotics and Automation* (1985), pp 500-505; E. Koch, C. Yeh, G. Hillel, A Meystel, and C. Isik, "Simulation of Path Planning for a System with Vision and Map Updating," *IEEE International Conference on Robotics and Automation* (1985), pp 146-160; D. K. Kuan, J. C. Zanisha, and R. A. Brooks, "Natural Decomposition of Free Space for Path Planning," *IEEE International Conference on Robotics and Automation* (1985), pp 560-570; T. Lozano-Perez and M. A. Wesley, "An Algorithm for Planning Collision-free Paths Among Polyhydrate Obstacles," *Communication of the ACCM*, Vol 22, No. 10 (1979), pp 560-570; E. K. Wong and K. S. Fu, "A Hierarchical-Orthogonal Space Approach to Collision-free Path Planning," *IEEE International Conference on Robotics and Automation* (1985), pp 506-511.

[4] E. Koch, et al., pp 146-160

[5] C.E. Thorpe and L.H. Matthies, "Path Relaxation: Path Planning for a Mobile Robot," *IEEE International Conference on Robotics and Automation* (1984), pp 576-581.

2. The path planning algorithm consists of two stages: static planning and dynamic planning.

3. The barriers/obstacles appearing along the path of movement are classified as either static or dynamic.

Static obstacles are those imposed on the design. A static obstacle may be created when a module bypass or an alley is closed. Information is gathered before the path is planned and not while the planned path is being executed. Most static obstacles could be overcome by the efficient use of the information provided in the landmarks and by communication with the central computer.

Dynamic obstacles are those which occur while the robot is moving along its planned path. Robot breakdown, items dropped in the pathway, and failure of the robot to access information from a landmark are examples of dynamic obstacles. When confronted with such a situation, the robot will plan an alternate path from its current position using the dynamic planning algorithm.

Nodal Representation of the Warehouse

Planning movement in a particular direction requires definition of specific points along the path. With this in mind, the warehouse modules with pathways were represented as a network model. The decomposition of the "group" in Figure 4 into a network structure is shown in Figure 6. Nodes in the network are points where two or more pathways intersect. For convenience, the nodes are numbered as follows. Nodes near the entrance of modules are identified by their corresponding block number and module number. The intermediate nodes (near the "exit" of modules) are identified by the two nodes on either side of the node under consideration. For example, A1 is the node near the entrance of module 1 in block A, while A1A3 is the intermediate node which has nodes A1 and A3 on either side. Because of this numbering system, the present warehouse structure has 96 nodes.

The nodes on the highway represent the junction. As seen in Figure 3, all the highway nodes are not equally spaced. For example, the distance between junction nodes A and AA and between D and DD, is almost three times the distance between the other junction nodes. Hence, four dummy nodes, two between junction A and AA and the other two between D and DD are introduced on the highway.

The relationship between the nodes is obtained by establishing a pointer for each node. The nodes to which the pointer points are called the successors of the node in question. Defining the parent-successor relationship for a nodal structure is important whenever a search is necessary.

Each node in the mainway is considered to have two successors and could be represented as:

```
N → (SUCC1, SUCC2)
```

Let the node N be at the intersection of i^{th} mainway and j^{th} alley. Moving along the i^{th} mainway in the direction of movement, the node on the $j+1^{th}$ alley forms the first successor, SUCC1. The second successor, SUCC2, is located on the $i+1^{th}$ mainway by moving in the prescribed direction along alley j. For example, consider the node A1A3 as shown in Figure 6. A1A3 is located on the mainway A and the alley AL2 of module 1 (in block A). Therefore the first successor of A1A3 (on the mainway of A and the alley adjacent to AL2 of module 1 in the allowed direction of movement) is A3. The second successor (along alley AL2, and on the mainway adjacent to A in the prescribed direction) is A10.

Thus node A1A3 is represented by

A1A3 → (A3, A10)

The entire warehouse is defined in terms of nodes and successor nodes. The junction nodes will have more than two successors while certain end nodes on the mainway will have only one successor. This is due to the constraints introduced in the movement.

The nodal structure of the warehouse is stored in the computer's memory and used in planning optimum paths. For easy searching and processing, the nodes are represented as a tree structure. An alternative way to represent the nodes is by a three(or higher)-dimensional matrix structure. The higher order matrices could be used to provide the various attributes required in the selection process.

Path Search Techniques

The following search techniques were considered.

Depth-First Search

The depth-first search dives deep into the search tree. While considering a node and its successors, this search chooses one successor, completely ignoring other alternatives at the same level, with the hope of reaching the destination using the original choice. The depth-first search is good when blind alleys do not get too deep. Otherwise the search will go on unchecked without realizing that the choice was a futile one.

Breadth-First Search

A breadth-first search pushes uniformly into the search tree. It looks for the goal node among all nodes at a given level before using the successors of those nodes to push on. Breadth-first is good when the number of alternatives at the choice points is not too large. In contrast to the depth-first search, it guarantees a solution if one exists. This search algorithm is explained in Appendix B.

Bi-Directional Search

This search proceeds in two directions: one from the start node and the other from the goal node. It automatically provides for alternate solutions when the two paths do not coincide and an optimal solution when they do coincide. Even though it is efficient, a bi-directional search is time consuming.

A Search*

This is a heuristic search which guarantees an optimal solution. At every level of the search, the nodes are selected on the basis of a heuristic evaluation function. However, the admissibility criterion always needs to be satisfied for the search to be fruitful.

Since the alternatives at a given level of the search are few (two alternatives for the mainway nodes and up to four alternatives for the highway nodes), the breadth-first search seems to be the best. A* search could be applied, but at certain points, it is difficult to satisfy the admissibility criterion. A bi-directional search was used by Katib[6] but the search creates problems when the two paths go on and on without meeting.

Path Planning Algorithm I

After designing the warehouse layout, the next step was to develop a simple algorithm to simulate and study the action of a single robot in the presence of static and dynamic constraints.

The algorithm consists of the static planning algorithm and the dynamic planning algorithm, and is based on the following assumptions:

1. The robot is cruising along the highway prior to receiving the command to visit a module. The path is planned from the junction nearest to the robot's position on the highway.

2. The path is selected on the basis of minimum time to reach the goal node from the start node.

3. The tree structure used for the search contains only those nodes within the module block selected. Hence, the planned path will not be affected by introducing constraints in the block that are not considered for the search.

Static Planning Algorithm

The static planning algorithm considers all the constraints that existed in the warehouse before the robot received an order. The command to visit a module is denoted by the block number (A,B,C, or D) and the module number (1-10). As soon as the robot receives an order, it moves to the nearest junction on the highway. The static path is planned from this junction.

Based on the block number of the target module, the algorithm selects the binary tree structure that contains the target module as one of its nodes. The allowed paths within the rows of modules belonging to adjacent blocks are included in the binary tree structure to permit flexibility in searching for a path to the target module. The tree is then updated by "breaking" the branches where constraints exist. The updated version of the binary tree is searched, breadth-first, to come up with two paths starting from two junctions, say B and CC (as explained in Appendix B).

[6] O. Katib, pp 500-505.

Based on the mean velocities on the highway and the mainway, the net time to reach the target module from the robot's current position is computed for each path. The path with the minimum time is selected.

Dynamic Planning Algorithm

For dynamic planning, the robot moves to the first node in its path and scans a distance of "D" ft; corresponding to the length of the next node in the path. If no obstacle is found, it moves to the next node and repeats the scanning process. (Note: the distance "D" = 60 ft if the next node is a module bypass or "D" = 120 ft if the next node is an alley.) Let an obstacle be found at a distance "d" ft from the robot. Based on the distance "d" of the path, the robot waits for a time period equal to (D-d)/VM (where VM is the mean velocity of the robot in the mainway) and repeats the scanning process. If the obstacle is found at the same position, the obstacle is considered static and that part of the path is declared closed. The dynamic planning algorithm then plans an alternative path to the target location from the robot's current position.

The tree is updated, breaking the links corresponding to the closed bypass/alley. The breadth-first search is conducted to arrive at an alternate path and the algorithm moves to the first step in the iteration. The example of path planning algorithm I given in Appendix C highlights the minimum time path selection process, the tree updating to take care of constraints, etc.

Path Planning Algorithm II

Algorithm II was developed to study the movement and interaction of two or more robots. This algorithm overcomes the restrictions posed by algorithm I. The various reasons for developing algorithm II could be summarized as follows:

1. Algorithm I was mainly developed to study the path selection and motion of a single robot. The path was selected on the basis of minimum time. However, it is advantageous for the robot to choose a minimum time path when it's idling and a shortest distance path when it's loaded.

2. The selection of a group for the search in algorithm I is equivalent to imposing the constraint that the robot can enter the mainway only through the two junctions of the group (refer to Appendix C). This may not always be the case in actual practice. The robot should have the freedom to choose any junction or node point so long as it satisfies the direction constraint and the optimality condition (minimum time path or shortest path). Also, there should not be any restrictions for the robot to plan its path starting from the nearest junction.

3. The two paths which algorithm I selected from within the group are actually the two best paths. But, when constraints are introduced in such a way that both paths are not directly accessible, algorithm I fails.

4. Algorithm I was found to have too severe a direction constraint imposed on the mainway and alley movements. For example, crossing the mainways was not permitted. It was therefore necessary to relax some of the constraints so that path planning became more flexible.

5. Only the nodes within the group selected are considered as candidate points for search in algorithm I. For an 8 x 5 module structure (warehouse layout) since there are only 10 modules and 4 junctions considered within the group, the tree structure is made of only 14 nodes. This limits the search area to a very narrow region.

6. The computations involved in algorithm I to arrive at the minimum time path are very detailed. The computation will tend to become more tedious when multiple robots are included.

Features of Algorithm II

- It takes into account the entire warehouse for the search. This results in a tree structure with 100 nodes for search. Note that the nodes to represent each module have additional "intermediate nodes" to allow for mainway crossings.

- Although the principle behind its static and dynamic planning is the same as that of algorithm I, the main difference between the two algorithms is in the scheme to arrive at the best path. When there are alternate paths from the start node to the goal, the breadth-first search algorithm always finds the path with the minimum number of nodes. Since the number of nodes characterizes the distance, this search will find the shortest path. In most cases tested, the shortest path corresponded to the minimum time path. This is believed to be due to the structurized layout and the strategy employed in depicting the tree structure.

- It allows the robot to define its start point at any of the nodes and need not always be at a junction node as required in algorithm I.

- It does not consider the search within a "group" based on the target module. Unless the constraints block the entrance into the target module, this algorithm can find a path to its goal.

- It has no direction constraints such as not allowing mainway crossings. However, the crossings are permitted only in certain specified directions. It takes into account the entire warehouse for the search, which results in a tree structure with 100 nodes for search.

- It facilitates computations since no detailed calculations are involved in the search process. This is advantageous when considering multiple robots. Appendix D illustrates the working of the algorithm II for a two-robot case.

Robot Movement Algorithm

Once the path has been selected, the next step is to execute the robot's movement along its planned path. Robot travel is by successive scanning and moving to the next node of its path. It may happen that two or more robots have to travel along the same path or have certain common nodes in their paths. In the former case, there may be a need to set priority levels for each robot such that one with higher priority is allowed to move first while the lower priority robots wait in a queue. The latter case may result in robots arriving at a node at the same time and colliding. It is therefore important to set the arrival time at the nodes in the path for each robot. This results in a node-time chart for each robot.

The robot movement algorithm is essentially a collision-avoidance strategy that considers the possible interaction of the robot with other robots and eliminates possible interactions through the node-time chart for each robot. This algorithm is called into operation as soon as the static planning is completed, and becomes the first step in the dynamic planning. The calculated time in the node-time chart corresponds to the arrival time of the robot at each node, so that when the robot moves from one node to another, the difference between the robots' arrival times at the nodes and the distance between the robots are taken into account to set the velocity of travel for each robot. The algorithm for a two-robot case is given below. An example of how the algorithm works is given in Appendix E.

1. Starting with the paths chosen for the two robots, establish the node-time chart by calculating the time to reach the nodes based on the robots' allowable speeds.

2. Compare the node points in the paths.

3. If there are no identical nodes, then no collision is possible.

4. Exit the algorithm with the node-time chart unchanged.

5. If there are identical nodes, compare the arrival times at the identical nodes.

6. If the arrival times are different, go to Step 4.

7. If the arrival times are within a tolerance limit specified, there is a chance of collision.

8. Depending on the priority set for the robots, modify the node-time chart of the lower priority robot so that the arrival time at the identical node is increased, thereby making the robot move slower than the allowed speed. Changing the arrival time at a node will automatically alter the arrival times at the subsequent nodes in the path.

9. Go to Step 2.

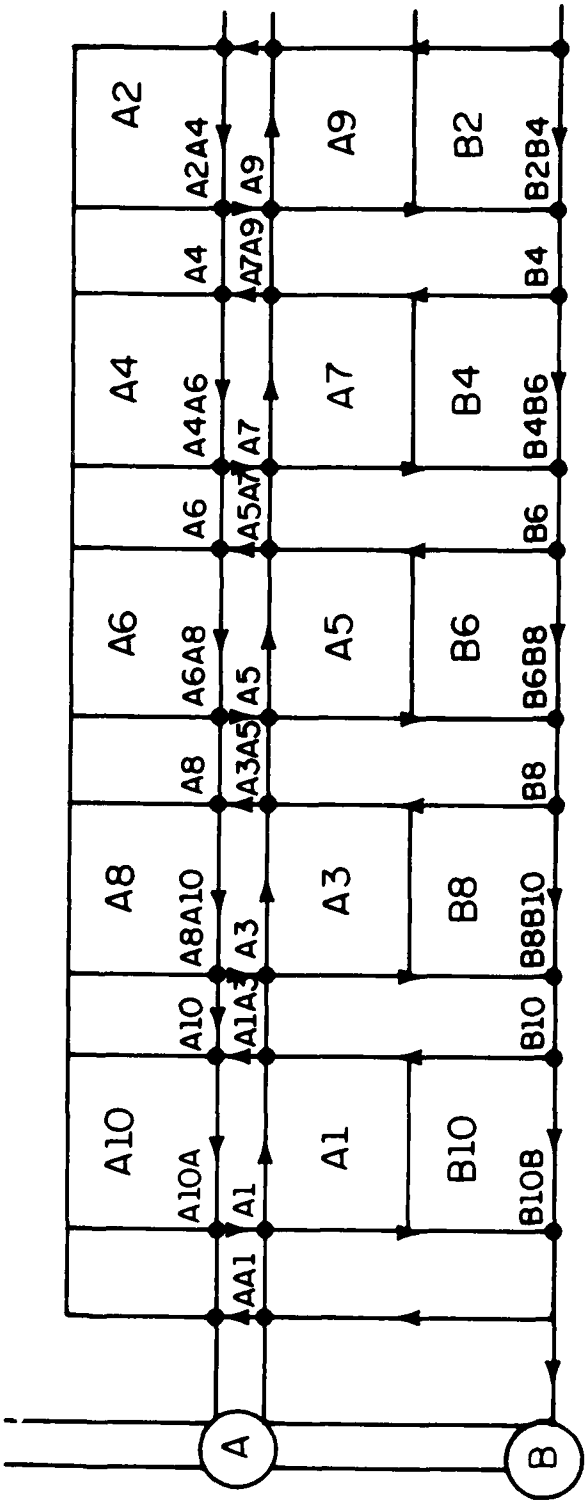


Figure 6. Nodal representation of a warehouse.

4 WAREHOUSE SIMULATION

An animated color graphics simulation model was developed to demonstrate the capabilities of the algorithms developed to run the warehouse activities. This representation provides the user with a clear picture of the working of the control algorithms. The user can study the layout of the warehouse material storage modules, the roadways, and the robots' movement as governed by the algorithm and directed by the user.

Figure 7 illustrates a block in the warehouse with its modules, highways, mainways, alleys, and junctions. Other blocks can be added on the top or bottom side of this block. The boxes numbered A10 to A2 through D1 to D9 are the storage modules. Number A10 represents module 10 of group A. Similarly, number D9 represents module 9 of group D. The outer squares (A, B, C, D, AA, BB, CC, and DD) are the junctions of mainways and highways. The outer two lines passing through the junctions are the highway enclosing the block. The horizontal lines within the block are mainways, and the vertical lines are alleys. The robots' movement is one-directional on highways, mainways, and alleys, as shown. Modules are shown as colored rectangular boxes on the simulation monitor. Junctions are shown as rectangles of different size and color. The roadways are shown with colored lines. Each robot is shown as a different-colored dot.

The numbers shown on the modules and junctions are used to define the starting and destination points for the robots. Alleys and bypasses are defined with respect to the modules.

Bypass A1 (Figure 7) is the roadway in front of module A1. Bypass A3 is similarly defined. The small path connecting the A1 and A3 bypasses is defined as path A1A3.

The alleys are defined by the direction of motion and module. Alley B1 is defined by the downward going vertical path to the left of module B1 and alley C10 is defined by the upward going vertical path to the right of module C10.

The simulation is interactive, allowing the user to introduce static and dynamic constraints. Static constraints are the disturbances present on the roadways before the robots start to move (e.g., broken-down robots, a fallen pallet of material, or a closed road). The static constraints are introduced before the robots select their paths but can be removed at any time. Dynamic constraints are the disturbances which occur while the robot is in motion (e.g., a stopped robot or a slipped pallet).

The capability of introducing static constraints also allows the user to convert the warehouse according to specific needs. For example, if the user defines a warehouse with the modules C7, C9, D4, D2 missing, this warehouse can be generated by blocking the mainways and alleys of C7, C9, D4, and D2 modules as shown in Figure 8. A large "X" represents a constraint.

The constraint for a path is defined with respect to the module bypass or alley. In Figure 8, constraint (1) is defined as alley A1, constraint (2) is defined as bypass C1, and constraint (3) is defined as bypass B3B5.

The simulation is done in three phases. During phase 1 the user inputs the starting and destination points for the robots, and any priority for the robots. If no priority is given, it is assumed that robot 1 has the priority. In phase 2, the static simulation, the user inputs constraints on the pathways. The robot chose a path based on these constraints. Robot motion is shown in phase 3, the dynamic simulation. While moving, the robot scans the road ahead of it and the user is given a choice to introduce or remove constraints. The robot will discover new obstacles and replan as needed to reach the target destination.

Figures 9 through 12 show two examples of simulation. In both examples, the starting point of the robot 1 is junction BB and the destination is module B10. The starting point of robot 2 is junction D and destination is module D9.

In the first example, Figure 9 shows static simulation of the system. No constraints are put in the paths of robots 1 and 2 and the robots select the following paths.

Robot 1 → BB → B2 → B4 → B6 → B8 → B10

Robot 2 → D → D1 → D3 → D5 → D7 → D9

Figure 10 shows the dynamic simulation of these robots. No constraints are added during dynamic simulation.

In the second example, Figure 11 shows static simulation of the system. Constraint's are put in the bypasses of B8 and D7 and in the alley of A3. The constraints in the bypass of B8 blocks the direct path of robot 1 and the constraint in the bypass of D7 blocks the direct path of robot 2. These constraints force the robots to choose alternate best paths. The paths selected are as follows:

Robot 1 → BB → AA → A → B → B1 → B10

Robot 2 → D → DD → D2 → D9.

Figure 12 shows the dynamic simulation of the robots. No constraints are introduced during dynamic simulation.

Software

IBM Advanced Basic language was used in developing the software because it has graphical commands. The software is user friendly and works in an interactive way with user. Figure 13 shows the flow of the system. The system starts with a picture of the warehouse layout and prompts the user for necessary information. User input is followed by static and dynamic simulation. When the destinations are reached, the system ends. The system is simple to run. Because the software is still being modified, it is not yet available. It will be available in conjunction with a future report discussing the modifications.

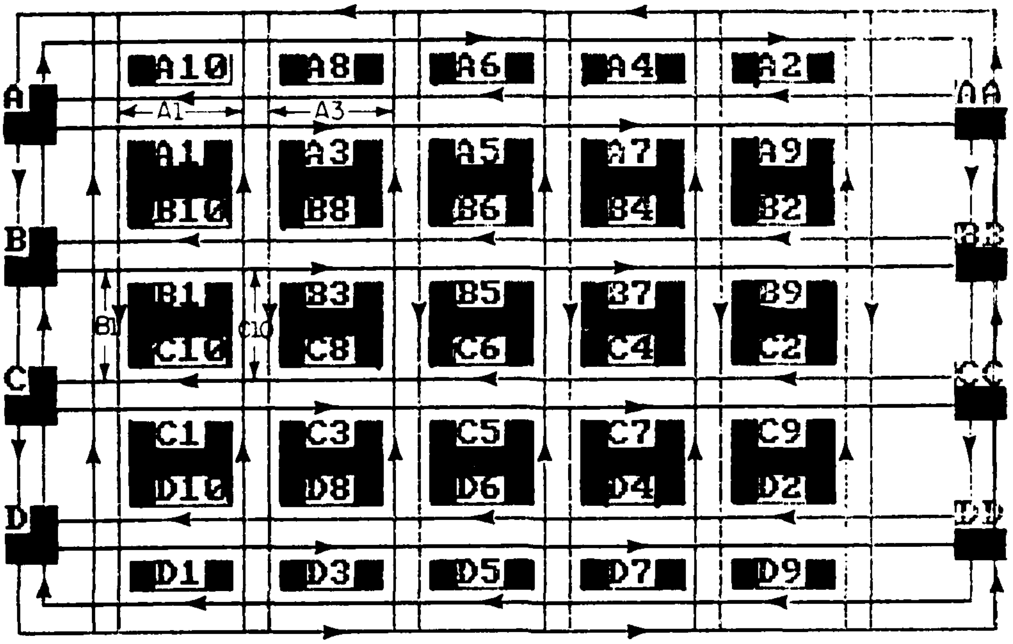
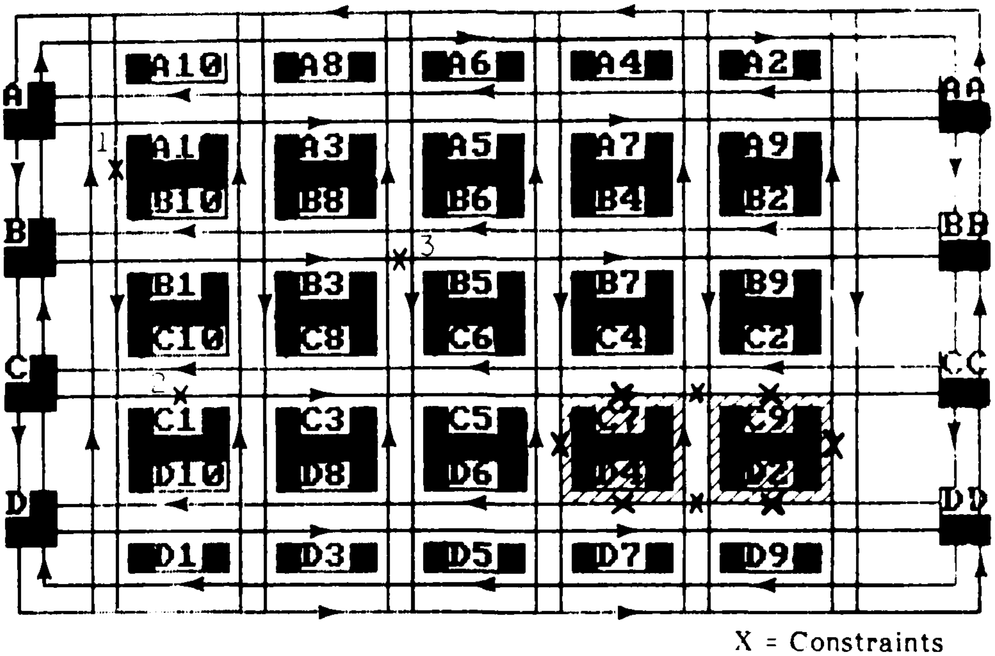


Figure 7. Simulation block layout showing bypasses and alleys.



X = Constraints

Figure 8. Static simulation showing constraints.

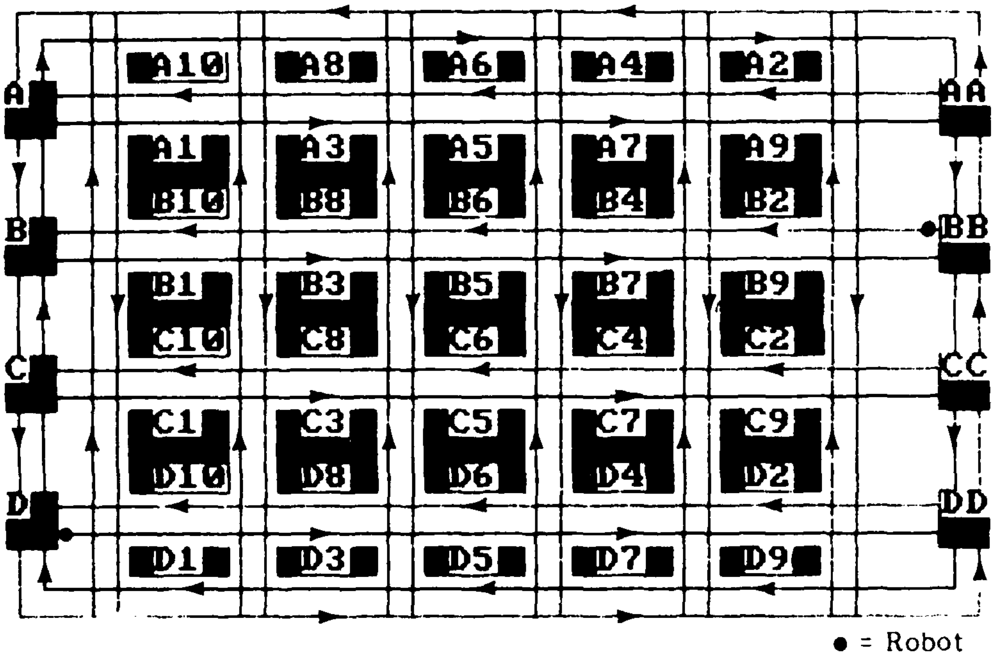


Figure 9. Static simulation without constraints.

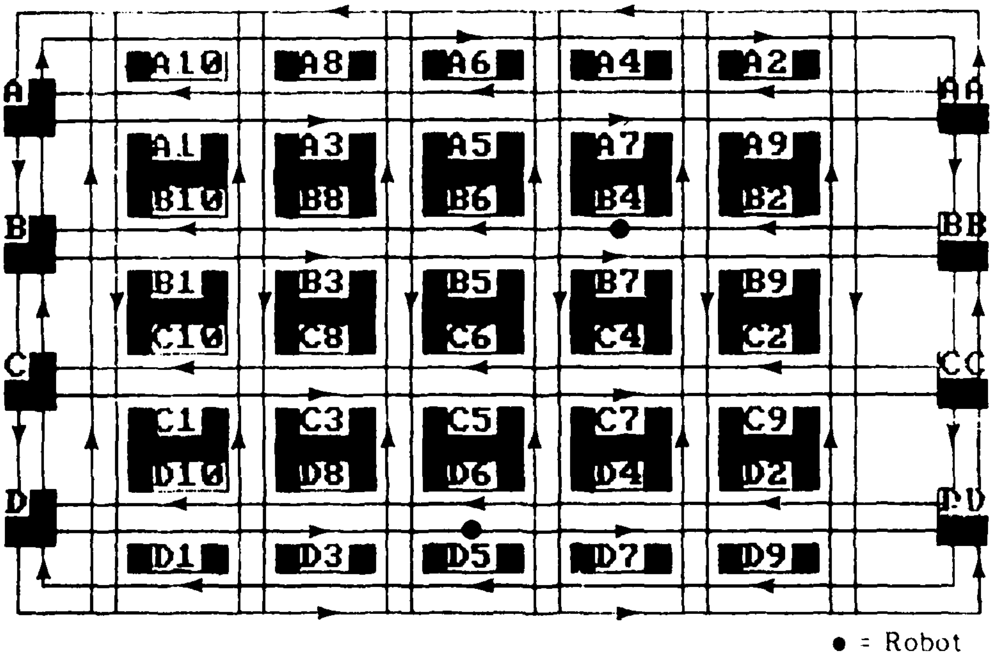
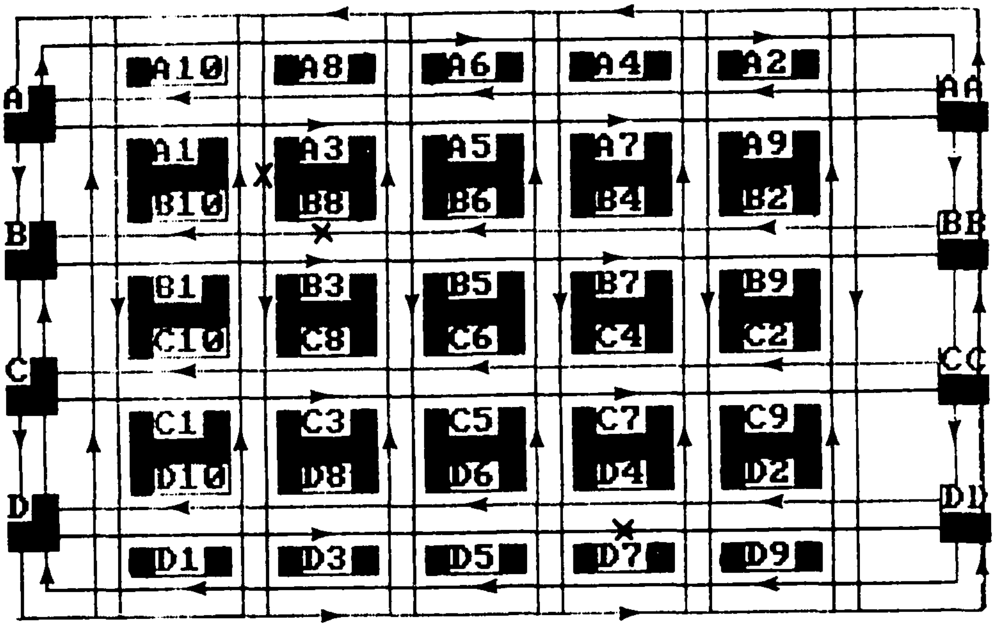
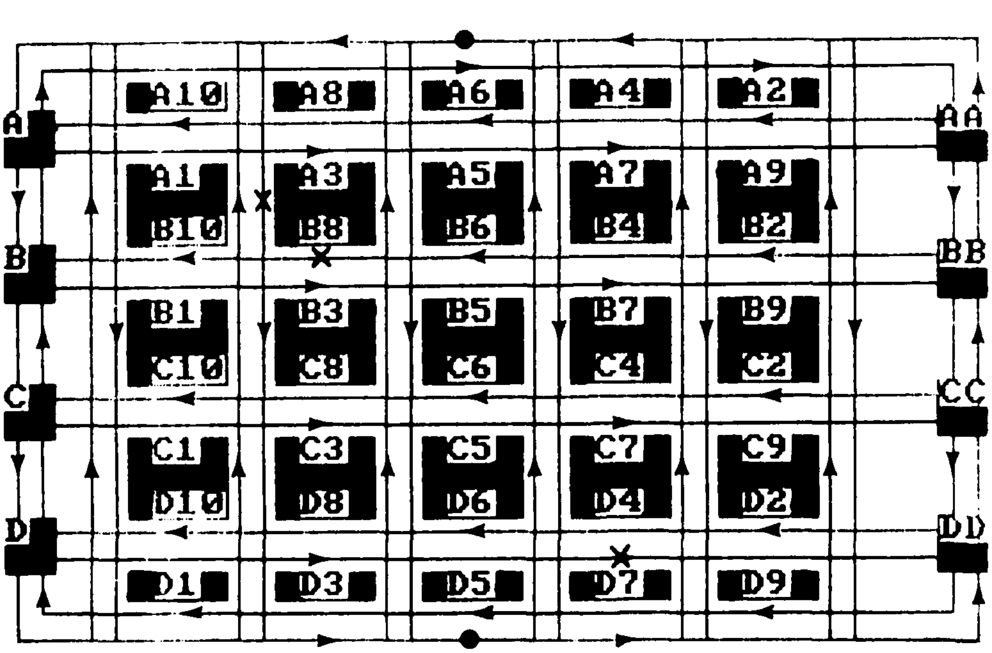


Figure 10. Dynamic simulation with no constraints.



X = Constraints

Figure 11. Static simulation with constraints.



X = Constraints
● = Robot

Figure 12. Dynamic simulation due to constraints.

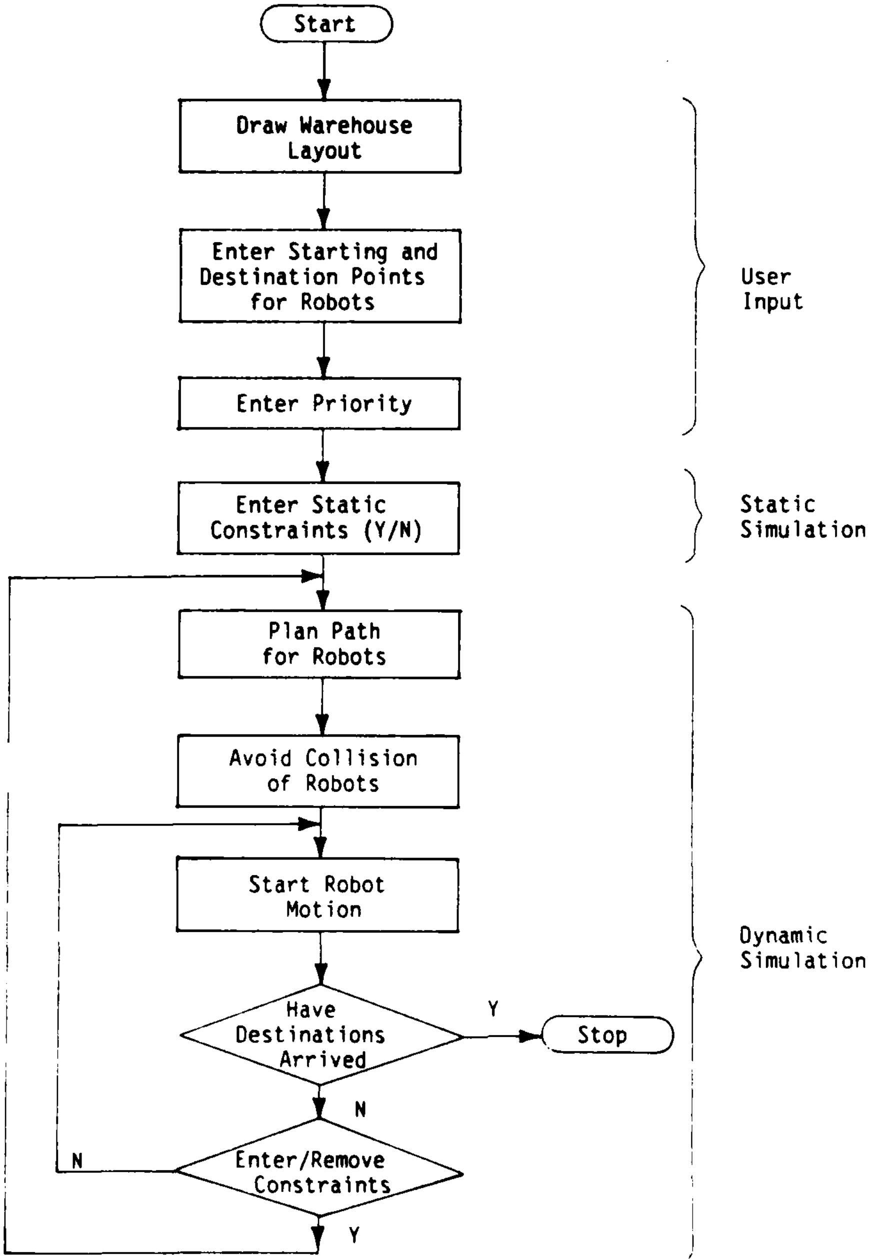


Figure 13. System flowchart.

5 SUMMARY

This research demonstrates that it is feasible to develop an intelligent guidance system for controlling mobile robots operating in temporary facilities such as construction warehouses.

The generality of the design is based on a method of representing the system of pathways in a network model, the use of landmarks for providing information to the mobile robots, and generalized algorithms.

The warehouse layout used in model development was modular and highly structured. It is therefore possible to define a warehouse as any group of modules, or blocks, in any arbitrary orientation as long as the nodes that connect the different blocks or modules are defined.

Modules, alleys, and junctions are physically identified by landmarks (e.g., patterns or bar codes mounted on the walls). Modules are uniquely identified in the appropriate landmarks. Landmarks eliminate the need for neat floor tapes to guide vehicles and for especially prepared floor finishes common to other types of Automated Ground Vehicle systems.

Algorithms were developed for (1) path planning, (2) collision avoidance. and (3) controlling the movement of the robots.

Path planning occurs in two places; initially upon receipt of a command to deliver or retrieve a storage item, and while on the path to the target module to avoid a discovered obstruction. The path planning algorithm uses the nodal structure of the layout to determine the quickest route to the target module for an unloaded vehicle and the shortest route to the designated destination for a loaded vehicle. After loading static constraints, a breadth-first search technique is used to define the best path. If an obstruction is discovered while the vechicle is following the planned path, the robot uses dynamic planning to plan a new route from its existing location to the target module. Discovery of an obstruction is communicated to the master planning computer and to all operating robots. Paths of operating robots will be modified simultaneously. There is hovever, a sacrifice in time required to search for new paths as dynamic constraints are encountered.

The collision avoidance and robot movement algorithms control operations in two ways. The collision avoidance system checks the next segment of the pathway for obstacles (either fixed or another robot). If an obstruction is identified, an alternative path is identified. Speeds of the robots are adjusted to avoid having two robots arrive at any particular point at the same time. Landmarks are used, in addition to location information, to provide data needed to ensure that the robots are maintaining proper orientation during movement, allowing the use of rough floor surfaces.

A simulation model of the whole system was developed using animated color graphic representation on an IBM-XT. The simulation progresses in three phases. In the first phase, the user inputs the required data on prompts. The second phase is the static path planning considering various pathways speeds, known obstacles, and coordinating plans with other operating robots. Phase three is the dynamic simulation that shows how robots move and how they select an alternate path in case of a dynamic constraint. This program was developed to demonstrate the control and planning algorithms. The program must be expanded to develop an operating system that can be easily implemented in the field.

6 RECOMMENDATIONS

The initial operating system should be approved as the first phase of a comprehensive warehouse automation system suitable for implementation in warehouses and storage yards with a wide variety of operating conditions.

Research should be continued to develop this initial system into a comprehensive operating system within a reasonable time. In addition to general program development, work should be invested in system failure mode identification and correction, path planning considering moving obstacles, and selection of sensors for collecting/transferring information.

In an automated material handling system, a failure occurs if a robot fails to accomplish an assigned task. For efficient system operations it is essential to know why the failure occurred, the extent of failure, and what actions should be taken. Failure analysis algorithms should be included in the system to address these questions. The path planning algorithms should also be modified to include appropriate responses to identified failures. The proposed path plan action may include finding an alternative path or suspension of activities in a certain module of the warehouse.

Path planning algorithms should be enhanced to optimize operations when dealing with a stopped or slow robot in the path of another robot. Methods to bypass such obstacles rather than using an alternative path would improve the system efficiency.

The search technique should be reevaluated to determine if another technique would improve the quality of the output path. Conditions such as an idling robot using a minimum time path and a loaded robot using the shortest path, may be more easily met by experimenting with heuristic functions and employing an A* type search instead of a breadth-first search.

Detailed requirements for the landmarks and the sensors to be mounted on the vehicles should be determined based on predicted operating conditions, safety features, and commercially available hardware and related software.

The graphic simulation package should be modified to include the system enhancements discussed above.

The general program should be expanded to simplify field implementation. This work will include a preprocessor that can convert a designed warehouse layout into a nodal structure and establish the relationships between the nodes and their successors.

After the program has been completed and the optimal landmarks and sensors are defined, the system should be tested and demonstrated at an Army warehouse using modified commercially available equipment. Decisions on field implementation and industrial involvement in producing system components should be based on an evaluation of the field test.

REFERENCES

Brooks, R. A., "Solving the Find Path Problem by Good Representation of Free Space," *International Journal of Robotics and Automation, IEEE* (1983), pp 190-197.

Chatila, R., and J. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots," Technical note LAAS-8905-9 (France, 1985) pp 138-145.

Crowley, J. L., "Navigation for an Intelligent Robot," *IEEE Journal of Robotics and Automation* (March 1985), pp 31-41.

Gouzenes, L., "Strategies for Solving Collision-free Trajectories Problems for Mobile and Manipulation Robots," *The International Journal of Robotics Research* Vol. 3, No. 4 (1984), pp 51-64.

Imai, H., and Y. Kawashima, "Advanced Automated Transportation System," Mitsubishi Electric Corporation, Japan, Proceedings of the First International Conference on Automotion and Warehousing, 1975.

Katib, O., "Real-Time Obstacle Avoidance for Manipulator and Mobile Robots," *IEEE* International Conference on Robotics and Automation (1985), pp 500-505.

Koch, E., C. Yeh, G. Hillel, A. Meystel, and C. Isik, "Simulation of Path Planning for a System with Vision and Map Updating," *IEEE* International Conference on Robotics and Automation (1985), pp 146-160.

Kuan, D. K., J. C. Zanisha, and R. A. Brooks, "Natural Decomposition of Free Space for Path Planning, "*IEEE*, International *Conference on* Robotics and Automation (1985), pp 560-570.

Lozano-Perez, T., and M. A. Wesley, "An Algorithm for Planning Collision-free Paths Among Polyhydrate Obstacles," *Communication of the ACCM*, Vol. 22, No. 10 (1979) pp 560-570.

Lu, S. C-Y., T. D. Brand, and S. G. Kapoor, *Sensor and Guidance Technology Related to Mobile Robot Devices*, Technical Report P-87/02 (USA-CERL, February 1985).

Miller, D., "A Spatial Representation System for Mobile Robots," *IEEE*, International Conference on Robotics and Automation (1985), pp 122-127.

Nilsson, N. J., *Principles of Artificial Intelligence*, Tioga Publishing Co. (1983).

Thorpe, C. E., and L. H. Matthies, "Path Relaxation: Path Planning for a Mobile Robot," *IEEE*, International Conference on Robotics and Automation (1984), pp 576-581.

Wong, E. K., and K. S. Fu, "A Hierarchical-Orthogonal Space Approach to Collision-free Path Planning," *IEEE* International Conference on Robotics and Automation (1985), pp 506-511.

APPENDIX A:

DESCRIPTION OF ROBOT VEHICLE

The Robot Vehicle resembles a cylinder with standard forks and lift mechanism. The maximum lift of the fork is 4 ft. The vehicle is mounted on a transport subsystem with four independently controlled wheels. The transport subsystem can be rotated independent of the remainder of the vehicle. Rotation is achieved by controlling the rotation of the wheels. The wheels are held flexible to the chassis so as to be put in contact with the floor surface securely at all times.

The control system has a main control unit (MCU) that communicates with the central control unit using radio signals, a recognition control unit (RCU) that identifies the sensor information performing the necessary computations and interacts with the MCU and the velocity control unit (VCU). The VCU provides the traveling and steering control.

The robot cruises at medium speeds along the mainways, slow speeds inside the modules, and at high speeds along highways.

APPENDIX B:

BREADTH-FIRST SEARCH ALGORITHM

The breadth-first search pushes uniformly into the search tree. The search looks for the goal node among all nodes at a given level before using the successors of those nodes to push on. The steps used to conduct breadth-first search include the following:

1. Create a search graph G, consisting solely of the start node, s. Put s on a list called OPEN.

2. Create a list called CLOSED that is initially empty.

3. If OPEN is empty, exit with failure.

4. Select the first node on OPEN, remove it from OPEN and put it on CLOSED. Call this node n.

5. If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n to s in G.

6. Expand node n, generating the successors, and install them as successors of n in G.

7. Establish a pointer to n from the successors of n that were not already in G (not already on either OPEN or CLOSED).

8. Add the successors of n to the end of the OPEN list. Reorder the list removing those nodes in the list of successors that were already in OPEN or CLOSED.

9. Go to Step 3.

The breadth-first search resulted in finding the shortest path from the start point to the target module. However, in most cases, the path corresponded to the minimum time path. The strategy employed in depicting the tree structure may be one of the reasons for meeting the minimum time criterion and is as follows:

The successor of each node is stored in the ascending order ot their arrival time. For example, consider the three cases shown in Figure B.1. For Case 1, nodes O and B are on the highway while node A is on the mainway. If the velocity in highway in tree times the velocity in mainways, then starting from O, the arrival time at node B is higher than that at A and hence the successors of node are listed as

$$O \rightarrow (B,A).$$

Case 2 shows the successor of node O the same distance from O and the pathway having the same velocities. Then the node O may be listed as list O → (A,B) or O → (B,A).

In Case 3, pathway OB is an alley and OA is a mainway. The velocities are the same, but the difference in distances will cause node A to be the preferred successor to node B.

$$O \rightarrow (A,B).$$

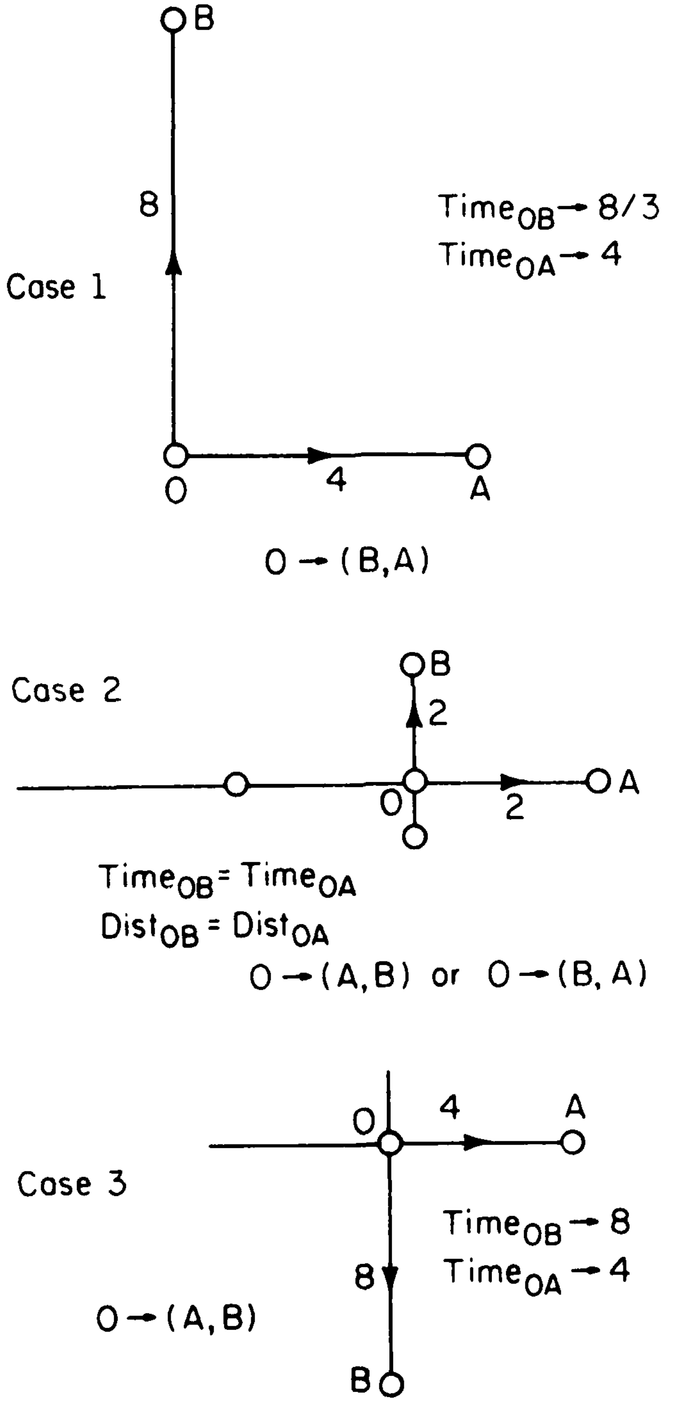

B
8
Case 1
$\text{Time}_{OB} \rightarrow 8/3$
$\text{Time}_{OA} \rightarrow 4$
O
4
A
O → (B,A)
Case 2
B
2
A
O
2
$\text{Time}_{OB} = \text{Time}_{OA}$
$\text{Dist}_{OB} = \text{Dist}_{OA}$
O → (A,B) or O → (B, A)
O
4
A
Case 3
$\text{Time}_{OB} \rightarrow 8$
8
$\text{Time}_{OA} \rightarrow 4$
O → (A,B)
B

Figure B.1 Successor selection.

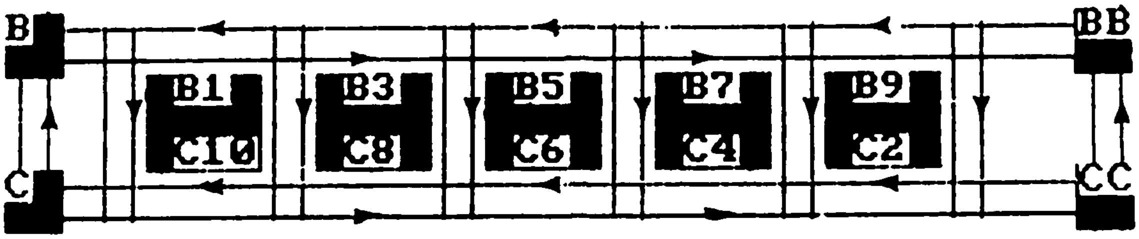
APPENDIX C:

EXAMPLE OF PATH PLANNING ALGORITHM I

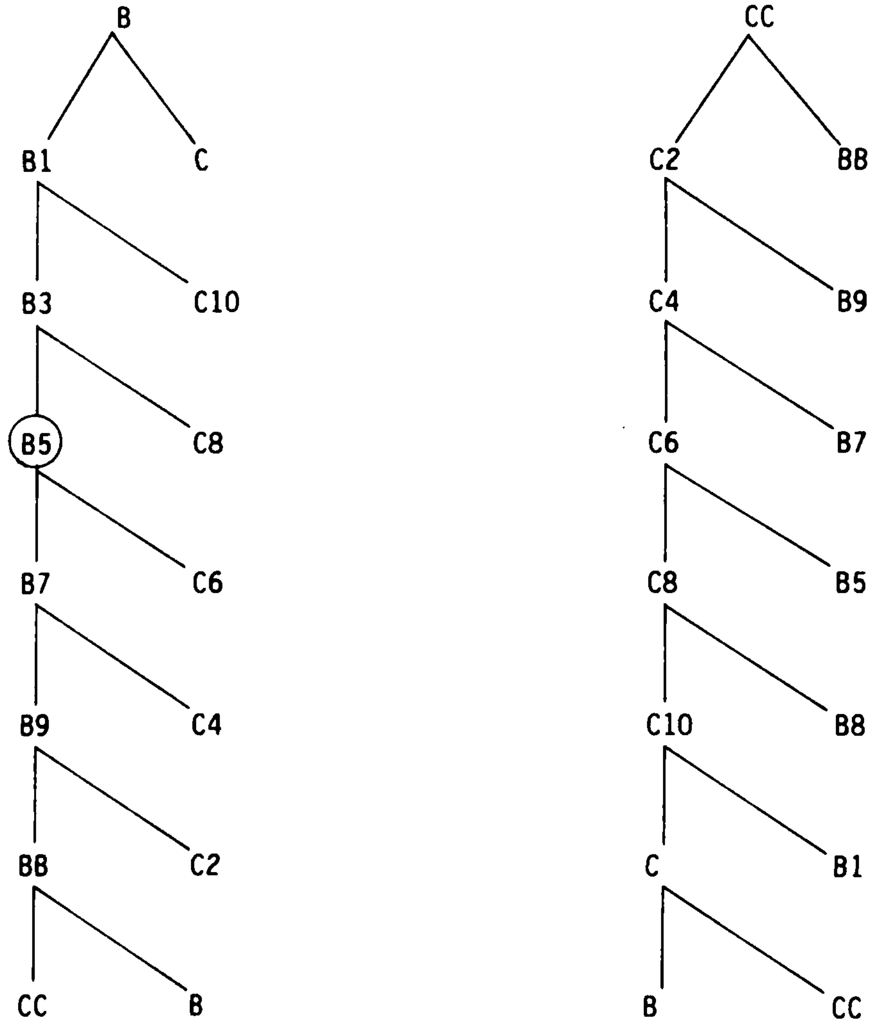
Static Planning

1. Target module to visit → B5

2. Select "group" based on the target number:
   Group selected = [B CC]



3. Generate tree from B and CC after introducing constraints.



4. Carry out breadth-first search from B to target B5 and from CC to target B5

Path 1: B → B1 → B3 → B5
Path 2: CC → C2 → C4 → C6 → B5

5. Enter robot's starting junction → A

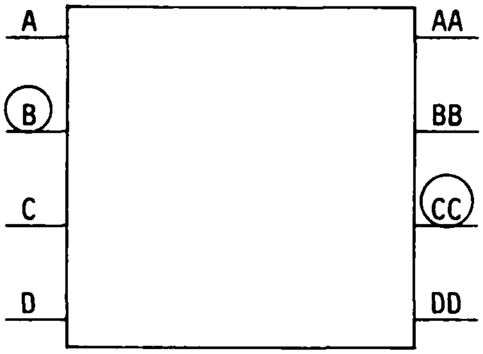
6. For path 1, based on highway velocity find path from

A to B based on minimum time.

Path =A → B.

For path 2, find path from A to CC based on minimum time.

Path: A → AA → BB → CC



7. The new paths from start point A to target B are:

Path 1: A → B → B1 → B3 → B5
Path 2: A → AA → BB → CC → C2 → C4 → C6 → B5

8. Based on highway and mainway velocities, select path with minimum time.

Path selected: A → B → B1 → B3 → B5

Dynamic Planning

1. Move from one node to other scanning for obstacles along the path

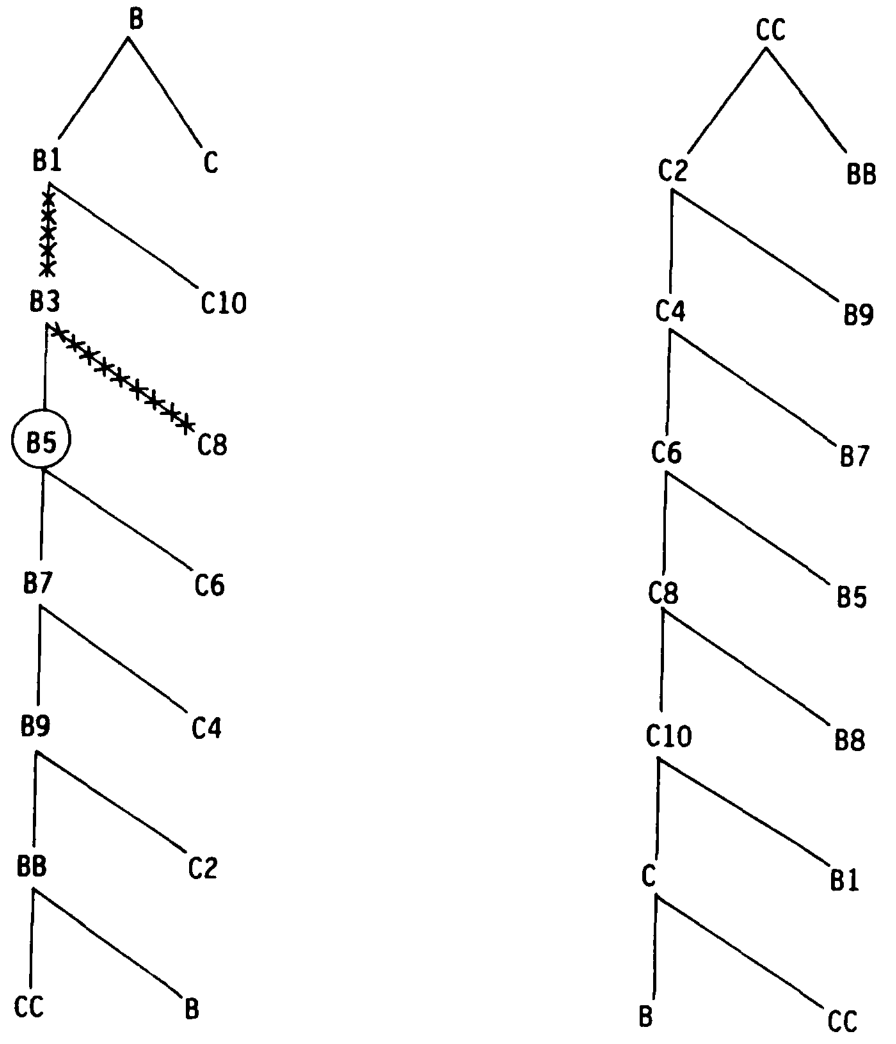
Move to B1
Scan from B1 to B3

2. Introduce dynamic constraints if any

→Block bypass of B3 and alley of B5

3. Update the tree structure

Updated tree after introducting constraints



xx Refers to Constraints

4. Carry out breadth-first search from B1.

Path selected = B1 → C10 → C → CC → C4 → C4 → C6 → B5

5. Go to step 1.

APPENDIX D:

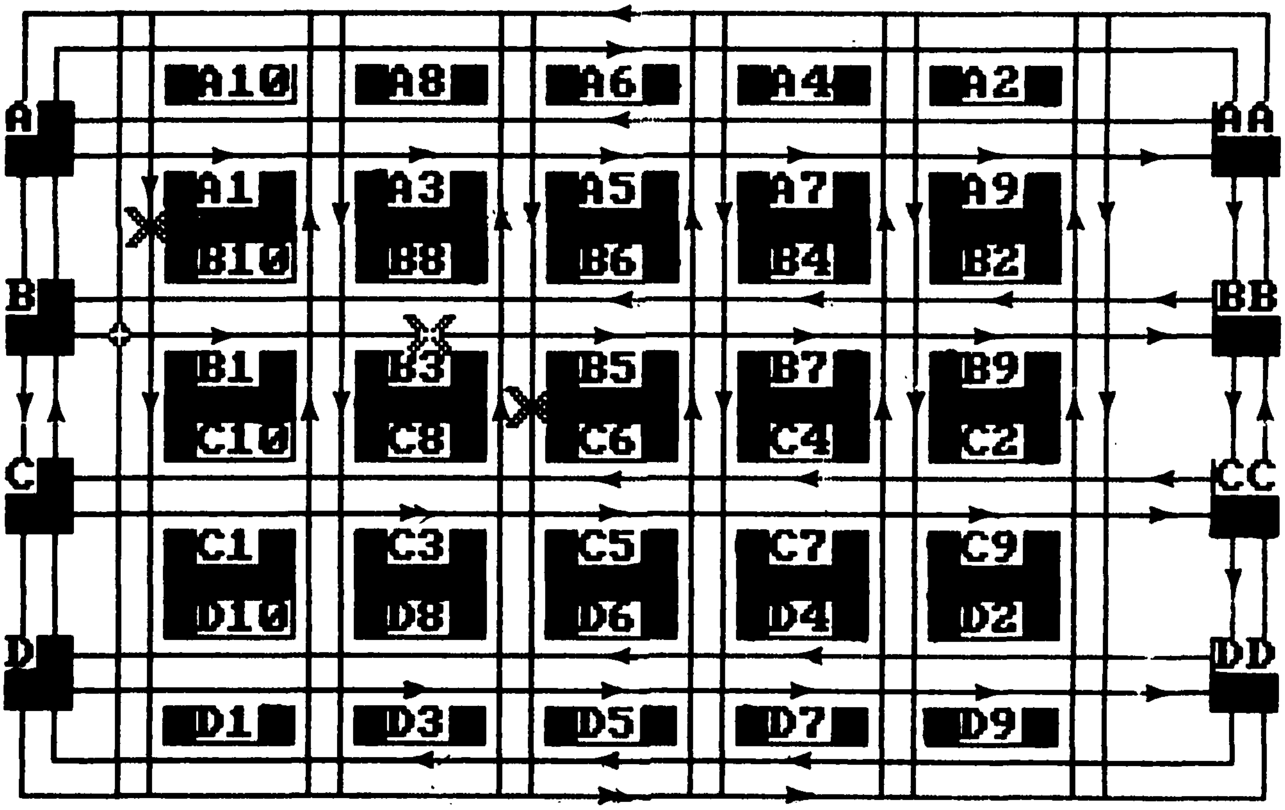
EXAMPLE OF PATH PLANNING ALGORITHM II

Static Planning

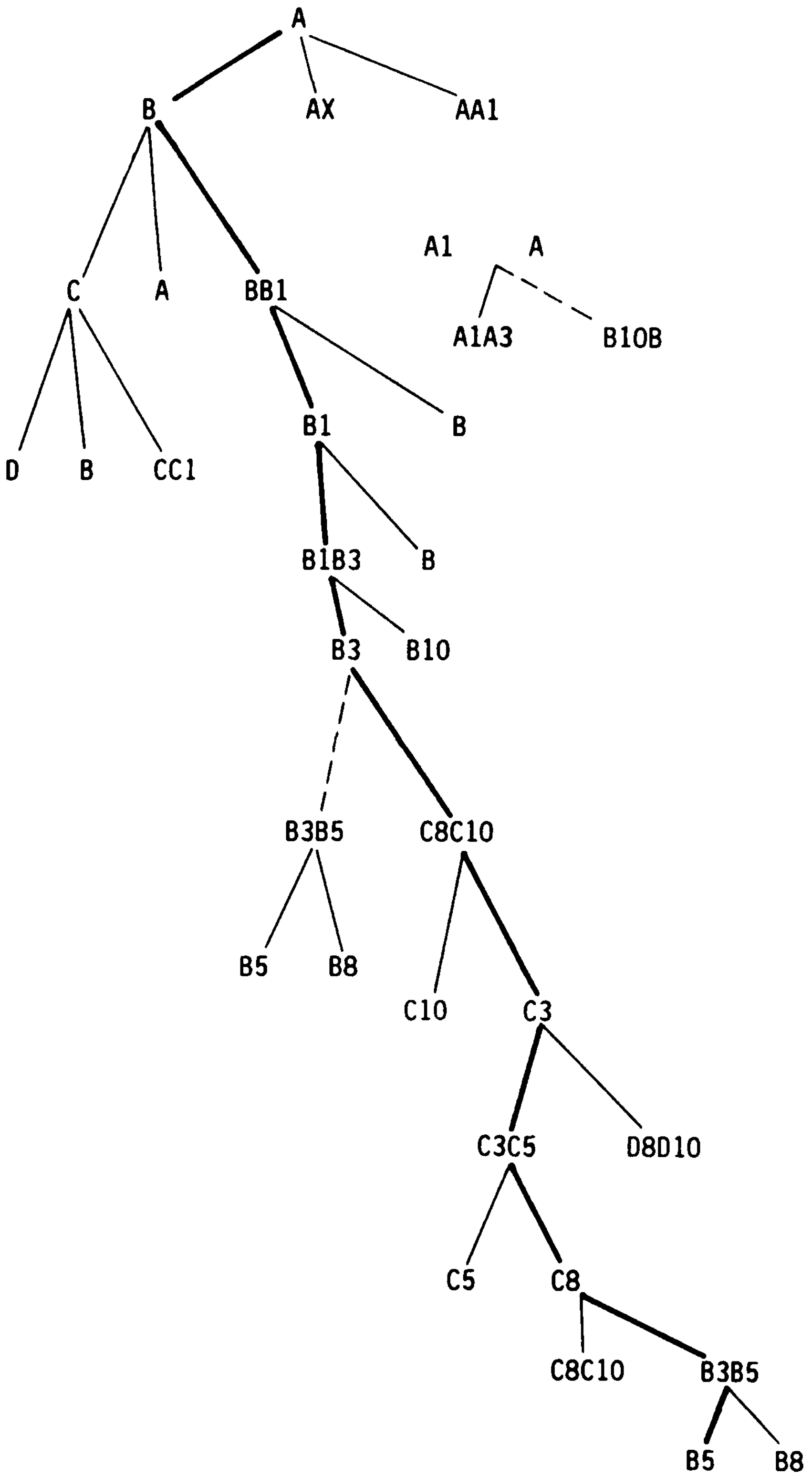
1. Enter Target for Robot 1 → B5
   Enter Start Point for Robot 1 → A
   Enter Target for Robot 2 → C3
   Enter Start Point for Robot 2 → A

2. Enter Static Constraints →

   Block Bypass B3,
   Block Alley B5,
   Block Alley A1



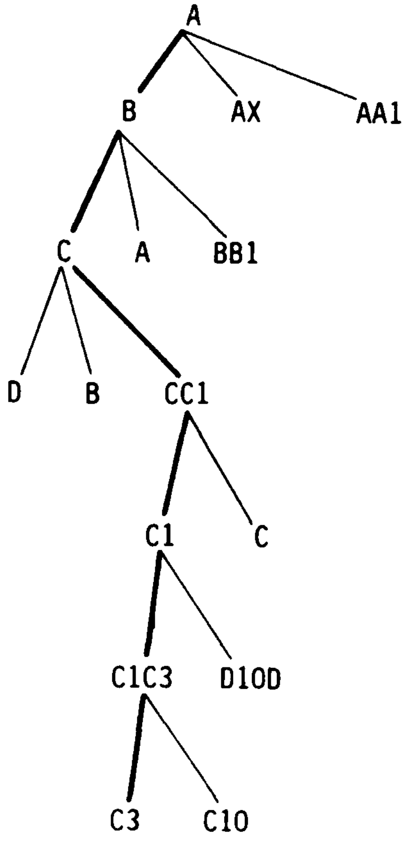
Path for Robot 1:



3. Path selected for Robot 1:

A → B → BB1 → B1 → B1B3 → B3 → C8C10 → C3 → C3C5 → C8 → B3B5 → B5

Path for Robot 2:



4. Path selected for Robot 2:

A → B → C → CC1 → C1 → C1C3 → C3

Dynamic Planning

The dynamic planning is similar to that explained for algorithm I. The later version of the algorithm has the capability fo interrupting the execution instead of moving from one node to other and scanning for the input of constraints.

APPENDIX E:

EXAMPLE OF ROBOT MOVEMENT ALGORITHM

1.

Path for ROBOT 1	Arrival Time
A	0
B	20
BB1	40
B1	48
B1B3	88
B3	96
B3B5	136
B5	144

Path for ROBOT 2	Arrival Time
A	0
B	20
C	40
CC1	60
C1	68
C1C3	108
C3	116

PRIORITY for Robot 1 = 0
PRIORITY for Robot 2 = 1

2. Compare paths for identical nodes

(other than start nodes)
1st identical node = B

3. Compare arrival times at the identical nodes

The arrival times are same
Collision is possible

4. Alter the arrival time of the low priority robot at the identical node

Change arrival time at B of ROBOT 1 to 26
Now the new file will look like this:

ROBOT 1		ROBOT 2	
Node	Arrival Time	Node	Arrival Time
A	0	A	0
B	26	B	20
BB	46	C	40
B1	54	CC1	60
B1B3	94	C1	68
B3	102	C1C3	108
B3B5	162	C3	116
B5	150		

There are no more identical nodes with the same arrival time. Hence, there is no possible collision. The robot velocities are adjusted depending on arrival times.

USA-CERL DISTRIBUTION

Chief of Engineers
ATTN: Tech Monitor
ATTN: DAEN-IMS-L (2)
ATTN: DAEN-CCP
ATTN: DAEN-CW
ATTN: DAEN-CWE
ATTN: DAEN-CWM-R
ATTN: DAEN-CWO
ATTN: DAEN-CWP
ATTN: DAEN-EC
ATTN: DAEN-ECC
ATTN: DAEN-ECE
ATTN: DAEN-ECR
ATTN: DAEN-RD
ATTN: DAEN-RDC
ATTN: DAEN-RDM
ATTN: DAEN-RM
ATTN: DAEN-ZCE
ATTN: DAEN-ZCF
ATTN: DAEN-ZCI
ATTN: DAEN-ZCM
ATTN: DAEN-ZCZ

FESA, ATTN: Library 22060
ATTN: DET III 79906

US Army Engineer Districts
ATTN: Library (41)

US Army Engineer Divisions
ATTN: Library (14)

US Army Europe
AEAEN-ODCS/Engr 09403
ISAE 09081
V Corps
ATTN: DEH (11)
VII Corps
ATTN: DEH (15)
21st Support Command
ATTN: DEH (12)
USA Berlin
ATTN: DEH (11)
USASETAF
ATTN: DEH (10)
Allied Command Europe (ACE)
ATTN: DEH (3)

8th USA, Korea (19)

ROK/US Combined Forces Command 96301
ATTN: EUSA-HHC-CFC/Engr

USA Japan (USARJ)
ATTN: AJEN-DEH 96343
ATTN: DEH-Honshu 96343
ATTN: DEH-Okinawa 96331

416th Engineer Command 60623
ATTN: Facilities Engineer

US Military Academy 10966
ATTN: Facilities Engineer
ATTN: Dept of Geography & Computer Science
ATTN: DSCPER/MAEN-A

AMMRC, ATTN DRXMR-WE 02172

USA ARRCOM 61299
ATTN: DRCIS-RI-I
ATTN: DRSAR-IS

AMC - Dir., Inst., & Servc
ATTN: DEH (23)

DLA ATTN: DLA-WI 22314

DNA ATTN: NADS 20305

FORSCOM
FORSCOM Engr, ATTN: AFEN-DEH
ATTN: DEH (23)

HSC
ATTN: HSLO-F 78234
ATTN: Facilities Engineer
Fitzsimons AMC 80240
Walter Reed AMC 20012

INSCOM - Ch, Instl. Div
ATTN: Facilities Engineer (3)

MDW, ATTN: DEH (3)

MTMC
ATTN: MTMC-SA 20315
ATTN: Facilities Engineer (3)

NARADCOM, ATTN: DRDNA-F 01760

TARCOM, Fac. Div. 48090

TRADOC
HQ, TRADOC, ATTN: ATEN-DEH
ATTN: DEH (19)

TSARCOM, ATTN: STSAS-F 63120

USACC, ATTN: Facilities Engr (2)

WESTCOM
ATTN: DEH, Ft. Shafter 96858
ATTN: APEN-IM

SHAPE 09055
ATTN: Surv. Section, CCB-OPS
Infrastructure Branch, LANDA

HQ USEUCOM 09128
ATTN: ECJ 4/7-LOE

FORT BELVOIR, VA 22060 (7)
ATTN: Canadian Liaison Officer
ATTN: British Liaison Officer
ATTN: Australian Liaison Officer
ATTN: French Liaison Officer
ATTN: German Liaison Officer
ATTN: Water Resources Support Ctr
ATTN: Engr Studies Center
ATTN: Engr Topographic Lab.
ATTN: ATZA-DTE-SU
ATTN: ATZA-DTE-EM
ATTN: R&D Command

CRREL, ATTN: Library 03755

WES, ATTN: Library 39180

HQ, XVIII Airborn Corps
and Fort Bragg
ATTN: AFZA-FE-EE 28307

Area Engineer, AEDC-Area Office
Arnold Air Force Station, TN 37389

Chanute AFB, IL 61868
3345 CES/DE, Stop 27

Norton AFB, CA 92409
ATTN: AFRCE-MX/DEE

AFESC, Tyndall AFB, FL 32403

NAVFAC
ATTN: Engineering Command (7)
ATTN: Division Offices (6)
ATTN: Naval Public Works Center (9)
ATTN: Naval Civil Engr Lab. (3)

ATTN: Library, Code L08A NCEL 93043

Defense Technical Info. Center 22314
ATTN: DDA (2)

SETAF Engineer Design Office 09019

Engr Societies Library, NY 10017

Natl Guard Bureau Instl. Div 20310

US Govt Print Office 22304
Receiving Sect/Depository Copies (2)

US Army Env. Hygiene Agency
ATTN: HSHB-E 21010

National Bureau of Standards 20899

END

5-87

DTIC